PhD Dissertation

International Doctorate School in Information and Communication Technologies

DIT - University of Trento

TRAFFIC ENGINEERING IN DYNAMIC OPTICAL NETWORKS

Elio Salvadori

Advisor:

Prof. R. Battiti

Università degli Studi di Trento

February 2005

DIT-05-023

Abstract

Traffic Engineering (TE) refers to all the techniques a Service Provider employs to improve the efficiency and reliability of network operations. In IP over Optical (IPO) networks, traffic coming from upper layers is carried over the logical topology defined by the set of established lightpaths. Within this framework then, TE techniques allow to optimize the configuration of optical resources with respect to an highly dynamic traffic demand. TE can be performed with two main methods: if the demand is known only in terms of an aggregated traffic matrix, the problem of automatically updating the configuration of an optical network to accommodate traffic changes is called *Virtual Topology Reconfiguration* (VTR). If instead the traffic demand is known in terms of data-level connection requests with sub-wavelength granularity, arriving dynamically from some source node to any destination node, the problem is called *Dynamic Traffic Grooming* (DTG).

In this dissertation new VTR algorithms for load balancing in optical networks based on Local Search (LS) techniques are presented. The main advantage of using LS is the minimization of network disruption, since the reconfiguration involves only a small part of the network. A comparison between the proposed schemes and the optimal solutions found via an ILP solver shows calculation time savings for comparable results of network congestion. A similar load balancing technique has been applied to alleviate congestion in an MPLS network, based on the efficient rerouting of Label-Switched Paths (LSP) from the most congested links to allow a better usage of network resources.

Many algorithms have been developed to deal with DTG in IPO networks, where most of the attention is focused on optimizing the physical resources utilization by considering specific constraints on the optical node architecture, while very few attention has been put so far on the Quality of Service (QoS) guarantees for the carried traffic. In this thesis a novel Traffic Engineering scheme is proposed to guarantee QoS from both the viewpoint of service differentiation *and* transmission quality.

Another contribution in this thesis is a formal framework for the definition of dynamic grooming policies in IPO networks. The framework is then specialized for an *overlay* architecture, where the control plane of the IP and optical level are separated, and no information is shared between the two. A family of grooming policies based on constraints on the number of hops and on the bandwidth sharing degree at the IP level is defined, and its performance analyzed in both regular and irregular topologies. While most of the literature on DTG problem implicitly considers the grooming of low-speed connections onto optical channels using a TDM approach, the proposed grooming policies are evaluated here by considering a realistic traffic model which consider a Dynamic Statistical Multiplexing (DSM) approach, i.e. a single wavelength channel is shared between multiple IP elastic traffic flows.

Keywords

Wavelength-routed networks, Virtual Topology Reconfiguration, Local Search, Traffic Grooming, Elastic Traffic, Generalized-Multi-Protocol Label Switching (G-MPLS), IP over Optical

Acknowledgement

La sera insegna ad attendere
il giorno che arriva come sempre
a chiudere i passaggi della notte.
F. Battiato, *Conforto alla vita*

I would like to thank my advisor prof. Roberto Battiti, for his precious advices and his assistance to my work. Many thanks to prof. Renato Lo Cigno, for his valuable support and for always being available for help and suggestions. My sincere thanks are due to Zoltan Zsóka for his collaboration and for his never ceasing patience in explaining the mysteries of GANCLES simulator. I also want to thank Dr. Mauro Brunato for the joined work on virtual topology reconfiguration. I am very thankful to Filippo Ardito and Mikalai Sabel who did great work in implementing some of the ideas presented in this thesis, and to Viet Thang Nguyen for reviewing this dissertation.

For creating a pleasant working atmosphere I want to thank all the present and past colleagues at the Computer Networks and Mobility (NetMob) Laboratory. I am also thankful to the many friends who have shared with me the moments of joy and encouraged me during the occasional bouts of frustration I have experienced during these three years.

Last but not least, I want to thank my parents, my sister and Nora for their great emotional support over all these years.

This material is based upon work supported by the Italian Ministry of Education and Research (MIUR) through the GRID.IT and ADONIS projects.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Since its introduction, Internet has dramatically modified ways of life and the way people work. This revolution has led to an unprecedented demand for worldwide data transport. At the same time, the quick evolution of optical technologies has allowed the transmission of huge amount of data on a single optical fiber. In this scenario, classical voice-oriented transport protocols used by telecommunication service providers have proved to be quite inefficient to run these emerging *next-generation optical networks*.

The goal of this chapter is to briefly introduce the context within which the thesis is positioned and to describe the problem considered for this research. The proposed solution and its innovative aspects are shortly described to highlight the main contribution of this dissertation.

1.1 The Context

The IP protocol and optical transmission techniques are going to play a fundamental role in the networking scenario of the next years — if not decades. In fact, most of the current intermediate network management layers (ATM, SDH, SONET, ...) should gradually disappear, leaving a scenario where IP packets are carried directly on high speed WDM-based optical connections. Optical packet switching and optical burst switching are long-term solutions for

the integration of optical transmission within a packet-based IP network; however, more traditional architectures, where packets are electronically switched in routers connected by circuit-switched optical connections, are going to dominate the commercial market for a long period.

GMPLS (Generalized MPLS) is also going to play its part in this scenario, enabling the use of Traffic Engineering (TE) techniques [15]. This modifies the IP behavior from pure datagram to "virtually connected," with the Label Switched Paths (LSPs) behaving as logical connections carrying highly dynamic IP traffic among routers. In this scenario, the interaction between routing and control of the circuit-switched optical network and that of the packet (or label) switched IP network is of the utmost importance for the end-to-end performance and the efficient use of network resources.

As stated in RFC 3717, in this emerging "IP over Optical" (IPO) network architecture, IP routers are attached to an optical core network and connected to other peers via dynamically established lightpaths [90]. The core network is composed of optical cross-connects (OXC), thus the optical layer is incapable of processing directly packets, bursts or any sub-wavelength capacity: it provides point-to-point connectivity between IP routers through fixed-bandwidth lightpaths. The set of all the lightpaths established over the physical topology is known as *virtual topology*.

1.2 The Problem

The pervasive usage of Internet services like World-wide-web (WWW), peer-to-peer and multimedia streaming has dramatically increased the volume of data traffic, which has overtaken that of traditional voice traffic. Furthermore in the next future a strong migration of real-time services over IP is foreseen. Future network infrastructures should therefore be able to accommodate increasing amount of data traffic, possibly with different Quality of Service (QoS) re-

quirements. The intrinsic difficulty in forecasting the behavior of IP traffic both at small and large scales [34] forces future network architectures to implement operation mechanisms which react efficiently to traffic changes.

Last, but not least, an important requirement for the future next-generation optical network is cost effectiveness. In spite of the evident importance of Internet traffic, today service providers' revenues are mainly coming from traditional voice services. After the so-called "optical bubble" in 2000 [39], service provider's new business models impose severe constraints on the network infrastructure upgrades. Today's networks must be designed and kept in operation by maximizing the exploitation of the available resources.

Traffic Engineering (TE) becomes therefore a fundamental solution to optimize the performance of a network and to control the network congestion through an efficient utilization of network resources. More specifically, the optimization may include the careful selection of paths for traffic flows through an appropriate route selection mechanism which takes into account traffic loads and network state, the re-routing of traffic flows to decrease network congestion and intelligent protocols which react efficiently to network failures.

In optical networks, the traffic coming from upper layers such as IP, ATM, MPLS or SONET/SDH is carried over the logical topology defined by the set of established lightpaths. Within this framework then, TE techniques allow to optimize the configuration of optical resources with respect to a particular traffic demand. When the network load changes dynamically, TE can be performed with two main methods, depending on the knowledge of the traffic demand and the level of integration between the data and the optical layers. If the demand is known only in terms of an aggregated traffic matrix, the problem of automatically updating the configuration of an optical network to accommodate traffic changes is called *Virtual Topology Reconfiguration*. If instead the traffic demand is known in terms of data-level connection requests with sub-wavelength granularity, arriving dynamically from some source node to any

destination node, the problem is called *Dynamic Traffic Grooming*.

1.3 The Solution

In this thesis we propose solutions for both the most relevant problems of Traffic Engineering in Dynamic Optical Networks: Virtual Topology Reconfiguration (VTR) and Dynamic Traffic Grooming (DTG).

Different methods are used to perform VTR in wavelength-routed networks, among them an emerging approach is based on *Local Search* techniques. The main advantage of such techniques is to keep the network disruption to its minimum since the modification involves only a small part of the network. We propose new load balancing algorithms for optical networks with arbitrary topology and based on IP-like routing, where every move in the Local Search modifies a single entry in the routing table of a node. A similar load balancing technique based on Local Search heuristics is presented in this thesis to reduce the congestion in an MPLS network. The key idea is to efficiently re-route Label-Switched Paths (LSP) from the most congested links in the network, in order to balance the overall links load and to allow a better use of network resources.

The Traffic Grooming problem has been proved to be NP-hard, thus only heuristics can provide sub-optimal solution when connection requests arrive dynamically. Although many algorithms have been developed to deal with this problem, little or no attention has been put so far on the QoS guarantees for the carried traffic, from the point of view of service differentiation *and* transmission quality. We first define a set of minimal QoS requirements for a High-Priority (HP) class of service at the IP level, and then we specify the corresponding QoS constraints at the optical level. Based on this differentiation, a novel Traffic Engineering scheme for IPO networks is proposed, based on two concurrent heuristics: a new DTG algorithm, which routes incoming connection requests to guarantee their QoS constraints in term of transmission quality, and a specific

preemption mechanism which provides service differentiation by minimizing the blocking probability for HP requests.

Another strong limitation of grooming algorithms proposed in literature is that they are normally studied with a very simple traffic model (circuit-like) that completely ignores the implicit interaction between the optical layer and the IP layer. In this thesis we first compare the performance of two simple grooming algorithms with a traditional, Poisson based traffic model and a more complex one that takes into account the IP traffic elasticity and the inherent interaction between the IP routing and the optical layer in *overlay* IP over Optical (IPO) networks. We show that ignoring the two layer interaction is not correct and may lead to wrong conclusions. A family of grooming policies based on constraints on the number of hops and bandwidth available at the virtual topology level is defined and analyzed in different regular and irregular topologies, discussing parameters setting and the impact of the number of available wavelengths per fiber on the grooming policy. Furthermore it is shown that dynamic grooming policies previously presented in the literature are particular cases of the family we defined, and that it is possible to define grooming parameters that lead to good performance regardless of the topology and that allow good scaling with the amount of optical resources.

1.4 Innovative Aspects

The proposed algorithms for Virtual Topology Reconfiguration extend the applicability of Local Search (LS) techniques to optical networks with arbitrary topology, while previous works based on LS were restricted to ring topologies only [52]. Furthermore a comparison between them and the optimal solutions found via an ILP solver shows calculation time savings for comparable results of network congestion. When applied to MPLS networks, the proposed load balancing algorithms based on a *reactive* mechanism show a reduced rejection

probability especially with long-lived and bandwidth consuming connection requests, thus proving a better network resource utilization compared to existing *preventive* constraint-based routing schemes, while guaranteeing a reduced computational complexity.

While a significant amount of research has been done on guaranteeing Quality of Service in pure IP-based networks, the problem of providing QoS guarantees to different services carried over high-capacity optical channels remains largely unsolved for wavelength-routed networks [59]. In the literature, the concept of "Quality of Service" in such networks assumes two main meanings: *service differentiation* or *transmission quality*, seldom jointly considered. The TE scheme proposed to guarantee QoS in IPO networks consider for the first time to the best of our knowledge both service differentiation and transmission quality. The proposed mechanism minimizes the network disruption, measured by the number of rerouted low-priority connections and new set-up lightpaths, and reduces the signaling complexity.

Another innovative result presented in this thesis is the introduction of a formal description of dynamic grooming policies, clearly defining the limits between grooming in *overlay* architectures and grooming in *peer* or *augmented* architectures [90], where there is total or partial integration of the optical and IP control planes. Furthermore, an analysis of the impact of realistic IP flows on the performance of grooming algorithms is performed. In fact, all the works on grooming algorithms presented in literature simply disregard the elastic nature of TCP/IP traffic, by modelling so-called "IP over WDM" like a traditional circuit switched traffic. In this thesis a simple analytical model and many simulation results are presented to highlight that the nature of IP traffic has a strong effect on the interaction between the two layers (data and optical) and that ignoring it is not correct and may lead to wrong conclusions. Within this framework the fairness of the grooming policy as a function of the flows length is also analyzed by identifying problems common to all strategies and proposing possible

solutions. Finally, to understand the impact of Traffic Engineering techniques in dynamic grooming, constraint-based routing techniques are introduced in the grooming policies and results are compared with a static, optimal grooming scheme, showing that dynamic grooming strategies with small complexity can perform nearly as well as an optimal one in static scenarios, yet preserving the adaptivity characteristics that allow higher performances in presence of unknown or variable traffic patterns.

1.5 Structure of the Thesis

Chapter 1 has introduced the context within which the thesis is positioned and the problem considered for this research. The proposed solution and its innovative aspects have been shortly described to highlight the main contribution of the dissertation.

Chapter 2 gives an overview of the current Optical Networking technology. A brief description of Routing and Wavelength Assignment is also given, a typical problem which characterizes optical networks. Furthermore, the concept of IP over Optical network (IPO) is discussed and a short introduction to MultiProtocol Label Switching (MPLS) and its implications to both IP and optical networks is described.

Chapter 3 describes the state of the art of the research related to Traffic Engineering (TE) methods for dynamic optical networks. In particular it gives an overview and a critical assessment of the most important works on Virtual Topology Reconfiguration (VTR) and Dynamic Traffic Grooming (DTG).

Chapter 4 presents new load balancing techniques for the VTR problem based on Local Search heuristic, which can be applied in optical networks where the forwarding mechanism is driven by the destination address only. The topic and partial results in this chapter have been discussed in the IFIP conference on *Networking* held in Pisa (Italy) in May 2002 [21], while its extension in-

cluding a comparison with an ILP formulation and a randomized scheme with reduced complexity has been published on the special issue on "*Dynamic Optical Networking: Around the Corner or Light Years Away?*" of *Optical Networks Magazine* of September/October 2003 [22].

Chapter 5 examines the performance of two new reactive TE mechanisms for congestion control in MPLS networks. Part of the work discussed in this chapter has been presented in the IEEE *Symposium on Computers and Communications* (ISCC) held in Antala (Turkey) in July 2003 [96], while detailed results are included in a technical report [99].

Chapter 6 presents a novel TE scheme for IPO networks to efficiently route sub-wavelength requests with different QoS requirements. The topic and partial results have been discussed in the 8th IFIP Working Conference on *Optical Network Design & Modelling* (ONDM) held in Gent (Belgium) in February 2004 [98], while new results have been presented in the IEEE *International Conference on Communications* (ICC) held in Paris (France) in June 2004 [97].

Chapter 7 describes a formal framework for the definition of DTG policies in IPO networks, and the performance of a family of grooming algorithms for overlay networks is analyzed by considering a realistic traffic model which consider a Dynamic Statistical Multiplexing (DSM) approach, i.e. a single wavelength channel is shared between multiple IP elastic traffic flows. Part of the work discussed in this chapter has been presented in the IEEE *GLOBECOM* conference held in Dallas (Texas, USA) in December 2004 [30], while a description of the GANCLES simulation software and of some results on the fairness of grooming policies have been presented in the 9th IEEE/IFIP Working Conference on *Optical Network Design & Modelling* (ONDM) held in Milan (Italy) in February 2005 [100]. The formal framework discussed in the chapter has been described in a paper submitted for publication [101].

Chapter 8 concludes this thesis, summarises contributions and achievements and outlines areas for further work.

Appendices provide a detailed description of the GANCLES simulation software and a guide to related papers produced during the period that the work for this thesis was carried out.

Chapter 2

An overview of Optical Networks

In recent years, advances in fiber optic technology have enabled the deployment of transmission systems which are capable of providing huge amounts of bandwidth across very long distances (e.g. intercontinental links). In particular, by using Wavelength Division Multiplexing (WDM), the fiber bandwidth can be divided into multiple wavelength channels, each operating at few Gigabit per second bit-rate (2.5 and 10 Gbps today, 40 Gbps in the near future). Currently available optical transmission systems can support over a hundred wavelength channels for an aggregate data-rate which can reach tens of Terabit per second over a single fiber.

At the same time, several drivers in the ICT market are forcing the traditional network protocol stack towards a strong simplification, where both IP and layer-2 Ethernet technology are gradually replacing the traditional IP over ATM/SDH redundant structure. The main advantage of this evolution is the elimination of the complex management problems in multi-layered networks. The emerging protocol stack is known as "IP-over-WDM" (or IP over Optical) [70], which indicates that IP packets are carried directly on high speed WDM-based optical connections. Some actors in the National Research & Education Networks (NREN) scenario have already started to provide channels strictly based on IP over WDM technology through their network infrastructure (e.g. CA*net4 [3]).

However further research is needed to guarantee that most of the functions provided by network layers such as ATM or SDH could be fully replaced by an IP-over-WDM only network infrastructure. In this chapter a brief overview of the optical networking technology considered throughout the thesis is provided in Sect. 2.1, while in Sect. 2.2 a brief description of Routing and Wavelength Assignment is also given, a typical problem which characterizes optical networks. Finally in Sect. 2.3 the concept of IP over Optical network is discussed, and a short introduction to MultiProtocol Label Switching and its implications to both IP and optical networks is given.

2.1 Optical network architecture

There is a wide consensus on the expected architecture of future optical networks, which are going to be built on the concept of *wavelength routing*. A wavelength-routed network is composed of optical cross-connects (OXC) connected by a set of fiber links according to some topology. The basic service provided by such an architecture is to provide a bandwidth-on-demand service by dynamically creating and tearing down logical connections (*lightpaths*) between client subnetworks. A lightpath is an optical channel which crosses the optical network without being converted into an electrical signal. This "transparency" is one of the greatest advantages of these networks: for example, it is possible to transmit on the same fiber channels carrying signals in different format and bit-rate, such as STM-16 and 10G-Ethernet.

An OXC simply provides the switching and routing functions to support the logical connections between network clients. In particular an OXC switches a signal carried on a specific wavelength from some input port to a different output port. In addition, some advanced OXCs allow to convert the input wavelength to a different output wavelength by using wavelength converters. Therefore an optical path (a lightpath) is set-up by properly setting the OXCs through-

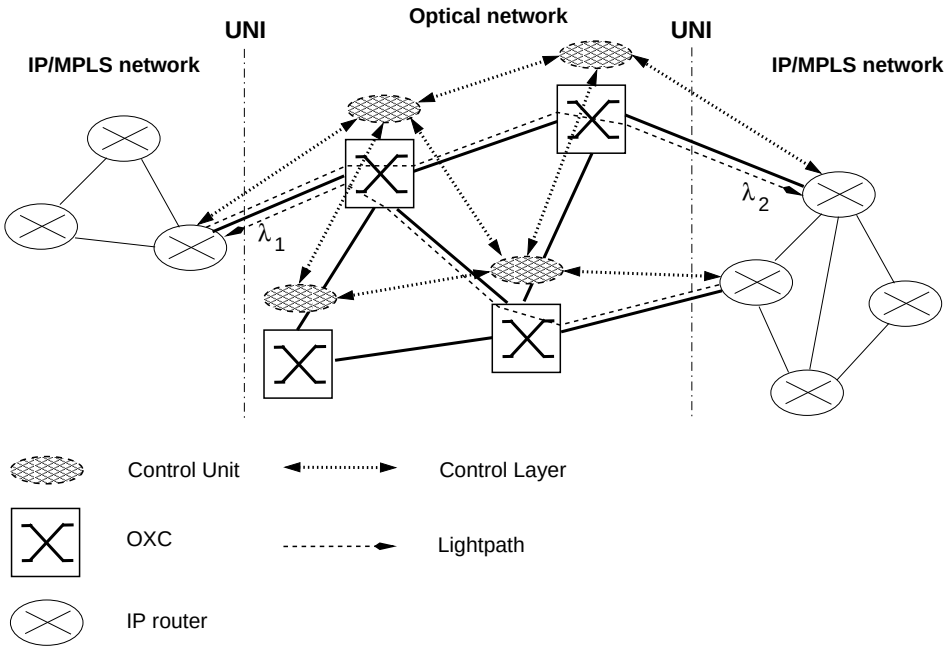


Figure 2.1: Optical network architecture

out the network path.

Fig. 2.1 shows a typical optical network architecture, where each OXC is managed by an electronic control unit responsible for the control functions related to the set-up or release of lightpaths. All control units communicate with each other through special control wavelengths set up between adjacent OXCs on each fiber link. It is possible therefore to distinguish between *control lightpaths*, which are strictly single-hop since they are electronically terminated at each control unit, and *data lightpaths*, which are originated and terminated at client IP/MPLS networks, therefore crossing transparently through the optical network.

Specific signalling protocols like G-MPLS (see Sect. 2.3.3) are running among OXCs through the so called *control layer*. These protocols are responsible for collecting and distributing relevant information regarding the network state. A number of standardization activities addressing the control plane aspects of next-generation optical networks are currently underway within different insti-

tutions, like the Internet Engineering Task Force (IETF) [7], the International Telecommunication Union (ITU) [6] and the Optical Internetworking Forum (OIF) [8].

As shown in Fig. 2.1, client subnetworks are attached to the optical core via edge nodes providing the interface between routers and optical cross-connects. This interface is denoted as UNI (user-to-network interface) and a first version has been already standardized by OIF [47]. Note that edge nodes perform as terminating points for the optical signal, converting it in electrical form to be routed in the IP/MPLS network. In Sect. 2.3 the assumption on the adoption of IP/MPLS protocols in these client networks will be explained in more detail. This hypothesis reveals the IP-centric nature of the emerging control architecture for the next-generation optical networks [116].

Even if not considered in the rest of this thesis, there are other two major solutions to transport packets through optical networks. In fact, the Optical Circuit Switching (OCS) solution analyzed so far is just one of the possible approaches to optical networking. While being one of the most promising from the technological point of view, this approach doesn't match entirely the highly dynamic nature of data traffic. For example, the signalling mechanism introduces long delays in the connection establishment which could not be easily tolerated by some of the incoming applications. Optical Packet Switching (OPS) proposes instead an integration between the IP and the WDM layer, by processing IP packets directly in the optical domain [43]. This can be obtained through optical packet switches whose architecture allows to read the packets header information and to switch them on a specific output interface. One of the main problem in OPS is the unavailability of optical RAM (Random Access Memory) to manage contention resolutions. Fiber delay lines, also known as *optical buffers*, can partially resolve such problem. In literature different optical switch architectures have been proposed to deal with this problem, by exploring all domains of an optical network: wavelength, space and time. Even though many

OPS testbeds have been already realized in different laboratories, the current technology is still quite far from solving most of the problems related to optical switches, therefore this approach can be considered as a long-term solution for future backbone networks.

Another promising approach is known as Optical Burst Switching (OBS), which can be seen as a compromise between the previous two (OCS and OPS) since it attempts to shift the computation and control complexity from the optical domain to the electrical domain. This technique was proposed in the early eighties and its application for optical networks was rediscovered successively [28]. In an OBS network, an edge node forms a burst containing a number of IP packets, and then sends a control packet along the network to configure each switch along the route through which the data burst will be transmitted. Two of the most investigated problems in OBS refer to burst assembly and burst reservation protocols, which are based on the trade-off between the burst size and the mechanism used for the path reservation. One of the main advantage of such technique compared to OPS is the elimination of optical buffers in core nodes: buffering functions are actually pushed to network edges. As a result, this approach can be considered as a mid-term solution for future backbone networks.

2.2 Routing and Wavelength Assignment

One of the fundamental problems in optical networks is how to select a route for lightpaths. As can be seen in Fig. 2.1, after a route for the lightpath is selected, it is necessary to select the wavelength on each of the fiber links which compose the path. Due to this strict coupling between route selection and wavelength assignment, this process is called Routing and Wavelength Assignment (RWA). The complexity of the RWA problem is due to two specific constraints which characterize an optical network: (i) the *wavelength continuity constraint*, i.e. a

lightpath must use the same wavelength on all the links along its route, and (ii) the *distinct wavelength constraint*, i.e. two or more lightpaths using the same fiber link must be allocated distinct wavelengths.

The wavelength continuity constraint may be relaxed if OXCs are equipped with wavelength converters [92]. This device converts the wavelength of an incoming optical signal to a different wavelength as the signal departs from its output port, while leaving the optical signal unmodified. A wavelength converter can be realized fully optically or through an electronic transponder, but in this case it violates the transparency which characterizes optical networks. OXCs equipped with such devices can introduce different levels of wavelength conversion capability (*full* or *limited*) according to the number of wavelengths which can be converted in each node. Wavelength conversion increases the routing choices for a given lightpath, therefore guaranteeing better performance. Note that when using full wavelength conversion, the RWA problem is reduced to the classical routing problem.

The main objective of an RWA algorithm is to exploit the available physical resources in an optical network (OXCs, converters, fibers links, number of wavelengths per fiber, etc.) in order to achieve the best possible performance. There are two main variants of the problem: the static and the dynamic RWA. While the *static RWA* emerges in the design and capacity planning phase of an optical network, the *dynamic RWA* arises during real-time operation phase and involves the dynamic provisioning of lightpaths. A brief overview of both these problems is given in the following subsections.

2.2.1 Static RWA

When the traffic demand is well-known in advance or traffic variations take place over long time periods, the problem of assigning routes and wavelengths to all the demands is known as Static RWA or Static Lightpath Establishment (SLE) problem. The objective is to assign routes to lightpaths in order to min-

imize the network resources usage (e.g. physical links and wavelengths). The SLE problem is known to be NP-complete [29], therefore polynomial-time algorithms which approximate solutions close to the optimal are preferred. A review of approaches to the SLE problem can be found in [120].

Different problems can be studied in this framework. The Physical Topology Design (PTD) refers to the phase when a network operator has a demand forecast and must decide a physical topology to connect its customers. This is a well-known problem in data-networks [18], while in wavelength-routed networks there are very few works about it [117]. The Virtual Topology Design (VTD) is instead the more classical research problem when considering SLE. The VTD deals with the problem of finding a set of long-lived lightpaths to be embedded into the physical topology, in order to realize a so called *logical* (or *virtual*) topology among client nodes. Note that fully-connected virtual topologies are difficult to realize since $N(N-1)$ lightpaths would be needed, with N being the number of edge nodes. In the rest of this section we focus on VTD only.

Static RWA can be solved as an optimization problem by using integer programming formulations, where the objective is to minimize the maximum congestion level in the network according to some specific resource constraints [93, 83]. The main advantage is that the problem is solved by considering the routing and the wavelength assignment problems jointly. However the disadvantage of ILP formulations is the reduced solution scalability, i.e. it guarantees the solution of very small problem instances (e.g. limited topology or reduced number of nodes). Another possible approach for solving the static RWA problem is its decomposition into four subproblems:

- *Topology subproblem*: which is the set of lightpaths (logical topology) to be imposed on the physical topology;

- *Lightpath Routing subproblem*: which are the routes for the lightpaths over

the physical topology;

- *Wavelength Assignment subproblem*: which is the wavelength to be assigned to each lightpath, according to the specific network restrictions;

- *Traffic Routing subproblem*: how to route data-traffic among edge nodes over the logical topology.

Note that this decomposition can lead to suboptimal solution for the static RWA problem, however it can lead to the formulation of effective heuristic algorithms.

Many algorithms have been proposed in the literature to deal with the static RWA problem, which can be subdivided into three main classes: (i) heuristics which solve the ILP problem suboptimally, (ii) heuristics which deal with a subset of the four discussed subproblems and (iii) heuristics whose main objective is to map regular logical topologies onto the physical topology.

A comparison among all the VTD algorithms proposed in literature is quite complex. In [41] the authors discuss bounds and theoretical considerations on the RWA problem, while in [67] a simulation-based comparison of the performance of these three classes of heuristic algorithms is analyzed. While all the cited papers refer to studies for point-to-point traffic, it is important to observe that other heuristics have been proposed when the considered traffic is multicast [95].

One of the most important variant of VTD is the Traffic Grooming (TG) problem. While in VTD the objective function is the minimization of the congestion level in the network, in TG the objective is how to groom a set of traffic flow requests in order to minimize network equipment or electronic switching cost [76]. In Chapter 3 this problem is discussed in more detail.

2.2.2 Dynamic RWA

In case of dynamic traffic demand, connection requests arrive at and depart from a network one by one in a random manner. Once established, the lightpaths remain for a finite period of time. When a new request arrives, a route and a wavelength need to be assigned to it with the objective of minimizing the number of rejections. According to the network state at the time a new request arrives, the available optical resources could or could not be sufficient to establish an optical connection between the source-destination nodes pair. An algorithm which finds a route and assigns a wavelength in a dynamic manner must be simpler compared to static RWA algorithms, therefore it is expected to perform poorly compared to them. The problem of lightpath establishment in a network with dynamic traffic demands is also known as the dynamic lightpath establishment (DLE) problem. A good review for this problem can be found in [121].

As noticed in the previous section, combining routing and wavelength assignment is a hard problem, therefore a classical approach to design DLE heuristic algorithm is to split RWA in two separate problems. Therefore most algorithms are based on the following steps:

- compute a certain number of candidate physical paths for each source-destination nodes pair and put them in a *path list*;
- insert all wavelengths in a *wavelength list* according to some order criteria;
- look for a feasible path and wavelength for the requested lightpath by picking them from the top of the corresponding list.

When considering the **route computation**, there are two main algorithms: the *static* routing algorithms, which calculate the paths independently from the network state, and the *adaptive* routing algorithms, which take into account the link state but require network nodes to exchange information. Another impor-

tant parameter is the number of path choices for the connection establishment. There are different possibilities, from the fixed routing to the fixed-alternate routing which compute k ($k > 1$) paths for all node pairs. Candidate paths are usually ordered according to some path cost, therefore it is possible to select paths with the larger number of free channels on the network links. When considering multi-cast traffic, more complex approaches are considered, such as [123]. A last important variation of this problem is the inclusion of physical layer impairments in the route selection. An optical signal may suffer from different transmission impairments during its propagation throughout the networks. Attenuation, chromatic dispersion, amplifier spontaneous emission (ASE), cross-talk, polarization mode dispersion (PMD) and various nonlinearities should be taken into consideration when choosing a physical path [108]. As discussed in [59], in order to provide QoS in wavelength-routed networks, it is mandatory to use an RWA algorithm that considers the QoS characteristics of different wavelength channels, such as [57, 11].

For what regards the **wavelength assignment** problem, there are different proposals for the way wavelengths are ordered in the corresponding *wavelength list*. There are two main possibilities, one is *static*, another *adaptive*. *First-fit* is one of the most used static methods; the idea is to exploit the usage of the wavelengths toward the top of the list in order to leave wavelengths toward the end of the list a higher probability of being available. *Most-used* and *Least-used* consider instead the dynamically varying usage of the wavelengths in the network, therefore they need nodes to exchange information about the wavelength status on each link.

DLE algorithms assume either *centralized* or *distributed* control for selecting routes and wavelengths. In the first case, a central controller is assumed to be available, such as in classical Telco networks, and responsible for the selection of the path and wavelength to be assigned to the incoming lightpath request. In the second case, up-to-date knowledge of the network state is not known to any

node therefore a distributed control may use pre-computed routes and search for free wavelengths on the links traversed. For scalability purposes, distributed protocols are highly desirable [121].

The performance of dynamic RWA algorithms is generally measured in terms of *blocking probability*, i.e., the probability that a lightpath could not be established due to lack of resources. The calculation of this performance index is extremely difficult even for very simple network topologies, due to the wavelength continuity constraint. Approximate analytical techniques have been developed in [109, 61]. Another important problem in WDM networks is the *fairness* between connections with different hop counts. In fact, longer-hop connections are less likely to be accepted than shorter-hop ones, and the situation becomes worse when a distributed control protocol is used, due to the increased possibility of wavelength reservation conflicts between simultaneous attempts to establish lightpaths.

Even if it is considered a variant of VTD, the Virtual Topology Reconfiguration (VTR) problem shall be included in the class of the dynamic routing algorithms in wavelength-routed networks, since it deals with traffic relations among source-destination nodes pair which change with time [52]. The idea is that a virtual topology designed for a specific traffic pattern could not be optimal when the pattern itself changes, therefore heuristics are needed to properly reconfigure the virtual topology to maximize the network performance. Reconfigurations requires that a few lightpaths in the existing virtual topology be removed and a few lightpaths be added to form a new virtual topology. There are a number of issues and different techniques have been proposed in literature to deal with them. In Chapter 3 this problem is discussed in more detail.

2.3 IP over Optical (IPO) networks

IP over WDM network is a model of transport network of growing interest. New services are continuously deployed over IP, and WDM evolution [50, 79] provides the transmission speed needed to pump the information through the network. Even Tier 1 providers have begun to encapsulate IP packets directly on the optical layer, avoiding the use of sophisticated middle layers such as ATM or SONET/SDH.

As discussed in Sect. 2.1, optical packet switching and optical burst switching are long-term solutions for the integration of optical transmission within a packet-based IP network. It is therefore supposed that classical optical circuit switching architectures, where packets are electronically switched in routers connected by switched optical connections, are going to dominate the optical networking scenario for a long period. In RFC 3717, IETF defines the framework for such network architectures, which are known as IP over Optical (IPO) networks [90].

In this section the architectural alternatives for interconnecting IP routers over optical networks are discussed in Sect. 2.3.1, while in Sect. 2.3.2 a brief description of MPLS scheme and of the related control protocols is given. The extension to G-MPLS is presented in Sect. 2.3.3.

2.3.1 A framework for IPO networks

Let us return to Fig. 2.1 which shows the way client subnetworks (IP/MPLS networks) attach to the optical core network made of OXCs. Two distinct planes are active over the user-network interface (UNI): the *data* and the *control* planes. Communication at the data (IP) layer can begin only if IP routers have established lightpaths in the optical network, therefore IP routers and OXCs must interact through the control plane in order to establish or release lightpaths for transferring IP packets. Even if it is not shown in the Figure, another

important interface in this architecture is the network-network interface (NNI), which defines the interaction between optical subnetworks belonging to different providers or consisting of switches from different vendors.

The assumption in this thesis is that similar control planes are used in the IP and optical networks. In particular it is assumed that a control plane based on IP *routing* protocols and MPLS *signalling* protocols is used in the optical network. Since control planes in the IP and optical networks can be loosely or tightly coupled, different interconnection models are possible for IPO networks:

- *Peer model.* In this model, the IP and optical networks are treated together as a single integrated network, where IP routers and OXCs are considered peer network elements, thus the topology and other network information are completely shared in a unified control plane. From the routing and signalling viewpoint, there is no distinction between the UNI and NNI, i.e., a single routing protocol instance runs over both the IP and optical domains.

- *Overlay model.* In this model, the IP and optical control planes are strictly separated therefore each domain runs its own routing and signalling instance protocols, and no information is shared among them (as in classical IP over ATM networks).

- *Augmented model.* This is an intermediate model characterized by separate routing instances in the IP and optical domains, but information from one control plane is passed through the other control plane (in general only from the optical to the IP layer).

The main advantage of the peer model is that it guarantees an optimized optical internetworking since the design and operation of the optical layer and the IP layers are integrated; its main drawback is that it raises security issues since routing information specific to optical networks needs to be known to IP routers. The augmented model partially solves these problems, but as for the

peer model, it can be applied mainly by ISPs which are also optical backbone providers [94]. In this model, the amount and kind of information which can be passed from the optical to the IP layer are still unclear [64], therefore both these solutions are not considered practical in the near term.

The main advantage of the overlay model is that it is the most practical for near-term deployment; an interface between the IP and optical level and dynamic lightpath capabilities in the optical level has been already proposed [47] and experimented in laboratories and research projects such as the european IST project LION [26]. Therefore this model can be applied by many providers, such as carrier's carrier and ISPs which lease optical infrastructure to optical backbone providers. Its main drawback is that the loss of integration between the two layers cannot guarantee the optimization of network resources [94].

These three interconnection models can be supported by three different routing approaches. In the overlay model each layer runs its own routing instances; an IP router can therefore trigger the set-up of new lightpaths through the UNI, but it would never know the corresponding path in the optical network, which is decided by some RWA performed in the WDM layer. In the peer model, a so-called *integrated routing* approach is active, since both layers run the same instance of an IP-based routing protocol, e.g. OSPF [80], with suitable "optical" extensions (e.g. wavelength occupancy, physical parameters,...). By using this information, a router could compute an end-to-end path to another router through the optical network, deciding which lightpaths need to be established and their routes. In the augmented model, the two routing instances are separated but some information are exchanged through some standard routing protocol, e.g. BGP [107], running between the domains. The reader is referred to RFC 3717 for more details [90].

Independently from the interconnection model, some IP-based control protocol is required to support automated provisioning of lightpaths within an optical network. The main mechanisms needed to guarantee correct operations in IPO

networks are the following: (i) neighbor discovery, (ii) link state update, (iii) route computation and (iv) path establishment. For example, once the route for a lightpath has been established according to some routing approach previously described, a signalling protocol must be invoked to set-up and manage the connection. Generalized Multi-Protocol Label Switching (G-MPLS) is emerging as the control-plane solution for this IP-centric network architecture.

2.3.2 Multiprotocol Label Switching (MPLS)

MPLS is an Internet Engineering Task Force (IETF)-specified framework to address problems as QoS management, scalability, service provisioning and traffic engineering, faced by today data networks. It was developed as a mean to introduce connection-oriented features in IP networks. In conventional routing, IP routers contain routing tables which are looked up using the IP header to decide how to forward the packet. The routing tables are populated by using IP routing protocols such as OSPF, IS-IS and BGP, which carry IP reachability information, therefore forwarding (data plane) and the routing table generation (control plane) are strictly coupled [86]. In MPLS, packets are forwarded based on short *labels* which move the route lookup for Layer 3 from software to high-speed hardware. Within the MPLS framework it is possible to demarcate the label-based forwarding plane from the routing protocol control plane. The main result of this demarcation is that MPLS provides better performance compared to current IP technology when considering key network applications such as Traffic Engineering (TE), Virtual Private Network (VPN) and traffic differentiation [35].

Label switching is an advanced form of packet forwarding that replaces conventional longest-prefix match forwarding with a more efficient *label-swapping* algorithm. Data are carried through an MPLS networks according to these steps: (i) label creation and distribution, (ii) creation of label information base at each router, (iii) creation of label switched paths (LSP), (iv) label insertion and ta-

ble look-up and (v) packet forwarding. Network devices participating in the MPLS framework are known as Label Switching Routers (LSRs) and Label Edge Routers (LERs). LSRs are high-speed switching routers supporting both the standard IP routing protocols and equipped with a label swapping forwarding component. LERs operate at the edges of an MPLS network and support multiple interfaces connected to dissimilar networks such as ATM, Frame Relay or Ethernet. These routers make classification and forwarding decisions by examining the IP header in the unlabelled packets, therefore they are responsible for the assignment or removal of labels as traffic enters or exits the MPLS network.

An important concept in MPLS is the forward equivalence class (FEC). A set of packets entering the network are grouped into a FEC and follow the same path in the network according to their address prefix and/or specific QoS requirements. The FEC to which a packet belongs is encoded as a fixed-length label, whose format (known as *shim header*) can be very different according to the underlying data-layer technology (Ethernet, ATM,...). Once a packet has been labelled, no further analysis of the packet header is performed in subsequent network hops. In particular, the label is used as an index into the label information base (LIB), which specifies the outgoing label and the next hop. The incoming label is therefore replaced with the new label and the packet is forwarded to its next hop. Normally, labels have local significance in the sense that they pertain only to hops between LSRs and are derived from the underlying layer-2 layers (e.g. VPI/VCI in ATM networks, DLCI in Frame Relay,...).

Label allocation in MPLS networks is performed by downstream peers (downstream with respect to routing). There are two basic mechanisms for label allocation: (i) downstream label allocation, in which the label assignments are made by the downstream node and distributed to neighboring LSRs, and (ii) downstream on-demand label allocation, in which the upstream LSR requests a label assignment from a downstream LSR. Assigning a label means that each

LSR creates an entry in the LIB after receiving a label binding.

The distribution of the label information to establish LSPs is guaranteed by specific signalling protocols. Label information in MPLS can be distributed in two main ways: (i) piggybacking label binding information on an existing routing protocol and (ii) using a new protocol, known as label distribution protocol (LDP), proposed by IETF. In MPLS two main signalling protocols are therefore used: the RSVP and its extension to traffic engineering (RSVP-TE) [17] and LDP with its extension for constraint-routing (CR-LDP) [9] for the establishment and maintenance of explicit routes. Both these protocols allows the implementation of constrained routing mechanisms in an MPLS networks.

In fact, when a new LSP must be set up prior to data transfer in an MPLS network, two possible routing approaches are possible: (i) hop-by-hop routing, i.e. as in an IP-based network, each router selects the next-hop router for a particular destination using the shortest-path algorithm, and (ii) explicit (or source) routing, where an ingress LSR (source) specifies the list of nodes through which the LSP traverses, and other LSRs simply obey the source's routing instructions. The routing algorithms used in this second approach are known as constraint-based routing (CBR) algorithm, and most of the time are derived from well-known QoS-based routing heuristic algorithms. Specific extensions have been proposed by IETF to most popular IP-based routing protocols, such as OSPF-TE and ISIS-TE, for the computation of constraint-based routes in an MPLS network.

Explicit routing can be used in an MPLS networks for a variety of reasons, such as to evenly distribute traffic among links by moving some of the traffic from highly utilized links to less utilized links (load balancing), create tunnels for MPLS-based VPNs, and introduce routes based on a quality-of-service criterion such as minimize the number of hops, minimize the total end-to-end delay, and maximize throughput. Label switching signalling protocols such as RSVP-TE or CR-LDP can be used to set-up these routes.

2.3.3 Generalized MPLS

G-MPLS [15, 14] extends the label switching framework proposed in MPLS to other types of *non-packet* based networks, such as SDH and wavelength-routed networks. In particular, the G-MPLS architecture supports the following types of switching: packet switching (IP, ATM, and frame relay), time slot switching for an SDH digital crossconnect (DXC), and wavelength, port or fiber switching in a wavelength-routed network.

A Generalized LSR (G-LSR) may therefore support up to five switch interfaces:

- a *packet* switch interface which recognizes packet boundaries and forward packets based on the content of the IP or the shim header;
- a *layer-2* switch interface that recognizes frame/cell boundaries and forward data based on the content of the frame/cell header (e.g. interfaces on Ethernet bridges which forward data based on the MAC header);
- a *time-division multiplex* interface which forwards data based on the data's time slot in a repeating cycle (frame) (e.g. interfaces on an SDH-based ADM or DXC, or PDH terminal multiplexer);
- a *lambda* switch interface that forwards the optical signal from an incoming wavelength (or waveband[1]) to an outgoing wavelength (waveband);
- a *fiber* switch interface which forwards signals from one (or more) incoming fibers to one (or more) outgoing fibers.

IETF is currently working on many Internet-drafts in order to extend the control protocols discussed in the previous section to support each of these interfaces[2]. New label forms are needed in G-MPLS to deal with both the

[1] A waveband is a group of usually contiguous wavelengths.

[2] The reader is referred to the web-page of the Common Control and Measurement Plane (ccamp) IETF working group for more details on this part [4]

time-division multiplexing and optical domains; inside a Generalized LSP (G-LSP) the label must guarantee not only the identification of packets, but also time-slots, wavelengths or fibers. In order to fix ideas, when considering a G-MPLS based optical network, a wavelength identifying an optical channel can be viewed as analogous to a label in MPLS [13]. In this scheme, OXCs can be seen as MPLS devices which aggregate packets (or smaller LSPs) from IP routers (LSR) into larger label-switched paths (G-LSP) associated with light-paths.

Inside the IPO framework, G-MPLS supports all the interconnection models analyzed in Sect. 2.3.1. The route of a G-LSP can be computed by using both explicit routing (usually implemented through some constraint-based routing algorithm which takes into account the network state) or hop-by-hop routing as in conventional IP networks. The routing protocols IS-IS and OSPF have been extended to advertise availability of optical resources (i.e., bandwidth on wavelengths, interface types) and other network attributes and constraints. Signalling protocols CR-LDP and RSVP-TE have both been extended to allow the establishment and release of lightpaths in G-MPLS, while a new link management protocol (LMP) has been developed to address issues related to the link management in optical networks.

Chapter 3

Traffic Engineering in Optical Networks

The main objective of Traffic Engineering (TE) is to improve the efficiency and reliability of network operations and management while optimizing network resource utilization and traffic performance, as stated in IETF RFC 2702 [12]. One of the most important motivations for TE is the ever increasing amount of highly dynamic IP data traffic with demands for assured QoS. Therefore telecommunication networks have to be flexible enough to react adequately to rapid traffic changes, in order to keep network congestion to reasonable levels and to make effective use of network resources.

TE solutions enables the fulfillment of all these requirements since they allow a network to choose routes for traffic flows while taking into account the amount of traffic load and the network state, or to move traffic toward less congested paths or to react to rapid traffic changes or network failures in short time intervals. In next-generation optical networks, these solutions can be adopted only by using an intelligent control plane such as G-MPLS, which is able to adequately handle network resources. TE in pure MPLS networks may include the careful creation of new Label Switched Paths (LSP) through an appropriate path selection mechanism (CBR), the re-routing of existing LSPs to decrease network congestion, the splitting of traffic between many parallel LSPs and protection schemes to guarantee timely restoration of traffic flows in case of link

or nodes failures. When considering IP over Optical networks, TE solutions refer to smart routing schemes for lightpath set-up, the intelligent reconfiguration of the virtual network connectivity (topology) to absorb traffic variations and the grooming of incoming data flows from either a circuit or packet bearer network layer onto high-capacity lightpaths, as well as to protection or restoration schemes for lightpath recovery.

The IETF RFC 3272 classifies TE schemes according to the following criteria [11]:

- **Response time scale**. It can be characterized as *long* when it refers to capacity upgrades of the network carried out in weeks-to-months time scale, *medium* (minutes, days) when it refers to response schemes relying on a measurement system monitoring traffic distribution and network resources utilization that subsequently provides feedback to online or offline traffic engineering mechanisms (e.g. to set-up or adjusting some LSPs in MPLS networks to route traffic trunks away from congested resources), *short* (picoseconds, seconds) when it refers to packet level processing function such as passive or active queue management systems (e.g. Random Early Detection - RED).

- **Reactive vs. preventive**. Reactive TE policies react to congestion problems initiating relevant actions to reduce them, while preventive policies prevent congestion based on estimates of future potential problems (e.g. distribution of the traffic in the network).

- **Supply side vs. demand side**. Supply side TE policies increase the capacity available to traffic demand in order to decrease congestion (e.g. balancing the traffic all over the network), while demand side TE policies control the traffic to alleviate congestion problems.

Furthermore, when considering IPO networks, another important differentiation criteria for TE schemes is the level of integration between the IP layer

and the WDM layer. Two are the main approaches to TE in this context [70]: **overlay** traffic engineering, if TE actions are effected separately in the IP layer and/or in the WDM layer and **integrated** traffic engineering, if these actions are coordinated across both layers. In the first approach optimization is pursued for one layer at a time, thus an optimal solution in a multi-dimensional space is found by sequentially searching different dimensions. The second is an emerging approach where optimization is pursued at both IP and WDM networks simultaneously in order to find a global optimal solution in a multi-dimensional space. Integrated TE faces enormous implementation complexity because the synchronization among a large number of optical nodes regarding the network state could take considerable time to converge. However, the overlay approach may not perform efficiently because the IP- and the optical-level Network Management Systems, which act separately, could create dangerous bottlenecks.

In this chapter an overview of three well-known TE mechanisms is considered and a brief discussion of the related state of the art is given, since new proposals for each of them are discussed in the rest of the thesis. In Sect. 3.1 the Virtual Topology Reconfiguration (VTR) problem in optical networks is discussed, while in Sect. 3.2 an overview of the most known TE schemes in MPLS networks is given. The Traffic Grooming (TG) problem is analyzed in Sect. 3.3, and its application to traffic engineering is studied in detail.

3.1 Virtual Topology Reconfiguration

As discussed in Sect. 2.2.1, the virtual topology is usually designed based on the estimated average traffic flow between the node pairs in a specific frame time. The length of this frame depends on whether the planning is short- or long-term. The traffic flow between the nodes is not constant and is subject to change with time, therefore the underlying virtual topology may not be optimal and reconfiguring it would help to maximize network performances. Recon-

figure the topology requires that a few lightpaths be removed and few added to form a new virtual topology. VTR can be therefore considered a *reactive* Traffic Engineering scheme, whose response time scale can typically vary from medium to long, according to the time interval considered to monitor the traffic changes on a virtual topology. Both overlay and integrated TE approaches are possible, according to the amount of information exchanged between the two layers.

There are a number of issues concerning VTR. Changing the current virtual topology will incur control overhead and traffic disruption. It may also require rerouting existing traffic (the so called: configuration migration), and this is perhaps the most dramatic option since the amount of information which can be carried on a single lightpaths is of the order of Gigabit per second. Therefore one of the main constraint of VTR is to minimize traffic disruption. Since traffic disruption depends on the number of lightpaths that are disturbed, it is highly desirable that the new virtual topology be as close to the current topology as possible.

A common hypothesis to all the VTR algorithms is that the traffic matrix specifying the traffic between each pair of nodes at a given time is known. Therefore all the heuristics proposed inside this framework consider the optimization of the virtual topology according to an aggregated traffic matrix, which is known at regular time intervals or when an important traffic change is detected (e.g., on the basis of some load threshold).

In the literature, there are three main approaches to the reconfiguration problem: direct, partial reconfiguration and local search [52]. In the following section a brief description of the main contributions in literature on VTR for arbitrary topologies is given.

3.1.1 Direct approach

This approach is characterized by the selection of new configuration independently of the current one, therefore the configuration migration subproblem is separated from the configuration selection subproblem, which is solved by using some classical VTD technique.

Wei [114] propose a scheme based on modifications of the WDM virtual topology to allow the IP/MPLS layer to accomodate additional load in the network. The algorithm is composed of two stages: in the first stage, the virtual topology is fixed, and MPLS-based TE mechanisms (constraint-based routing, see Sect. 3.2.1) are executed to balance the load across the network. When an LSP path cannot be established between a given ingress-egress pair due to the absence of spare capacity (network congested), a WDM reconfiguration algorithm derives a new optimized virtual topology based on the traffic demand matrix collected from the IP router measurements. Smaller average hop distance for the new virtual topology lead to reducing bandwidth consumption and expanding bottleneck capacity. The main limitations of this proposal is its limited applicability for a distributed implementation, that reduces its use as an off-line optimization solution. In fact, an on-line WDM reconfiguration of the topology could have a dramatic impact on the flowing traffic, leading to intensive packet retransmission and reordering.

A comparable method is proposed by Sato et al. [102]: the idea is to implement layer 3 functionalities at every optical node, in order to merge as many IP traffic streams as possible into the network wavelengths, thereby fully utilizing the wavelength bandwidth. The extensive use of wavelength converters in the network, the delays introduced by layer 3 processing at every node and the continuous reconfiguration of the WDM virtual topology according to the IP traffic demands are the main drawbacks of this approach.

3.1.2 Partial reconfiguration approach

In this section other VTR techniques are considered whose main objective is to reduce the degree of network disruption by sacrificing the optimality of the new virtual topology configuration.

Sreenath et al. [105] propose an heuristic called Path-Add whose aim is to improve the network configuration by reducing the average traffic-weighted hop count. The main idea is to consider K node pairs among which the traffic flows have the greatest traffic-weighted hop count, and join each pair with a single lightpath. If this lightpath cannot be established, up to L sub-sets of existing lightpaths are considered for deletion. The heuristic explore all possible options and selects the one which gives rise to a virtual topology with the smallest traffic-weighted average hop count. Even if the number of reconfiguration changes is bounded by another parameter D, this heuristic can potentially lead to the removal of many lightpaths, with dramatic impact on the traffic disruption.

Puype et al. [88] propose Multi-Layer Traffic Engineering (MTE) schemes based on two main strategies: a "reactive" one where MTE actions are triggered only by the detection of network congestion and a "proactive" one that tries to keep the network optimal at all times, triggering a reconfiguration whenever optimizations are possible. In the paper MTE is defined as a three step process: (i) how network congestion is detected, (ii) which decisions must be taken to improve the virtual topology and (iii) which actions have to be performed in the IP over WDM network in order to realize the desired modifications. The basic idea is to define two threshold values T_{high} and T_{low} indicating over- and underloaded links respectively; the load monitored on every network link is continuously kept between these two thresholds, in order to avoid congestion over the IP routers. The results of the proposed scheme, expressed only in term of Packet Loss Ratio over the IP layer, are not clear because the authors consider

only the application of their algorithm for a traffic load that rise from 80% to 160% of the forecast, over an unknown network topology.

A similar mechanism is proposed by Gençata et al. [49] through an heuristic adaptation algorithm used to evaluate the performance in term of number of added or removed lightpaths for specific values of high and low thresholds. In particular, the proposed algorithm reacts to imbalances caused by traffic fluctuations by promptly adding or deleting one lightpath at a time, thus liberating network resources in a less disruptive manner than the lightpath deletion in [105]. The authors show also a detailed study of the impact of the observation period considered for the link load measurements. All the experiments are performed over a network made of optical nodes with full wavelength-conversion capability. Moreover, the authors use "realistic" traffic in their experiments, modelling the load one can measure over a typical Internet backbone link. Being this traffic composed of aggregated traffic, the relative traffic matrix will never show unexpected sharp characteristics, thus putting the proposed algorithm in the most favorable conditions to perform optimally.

Iovanna et al. [56] propose a hybrid approach for the routing of IP/MPLS LSPs over a WDM layer which takes advantage of a combined use of off-line and on-line routing strategies to optimize the use of network resources. The proposed heuristic approach is composed of two main phases: (i) an initial paths set-up (performed off-line) by means of a successive shortest-path algorithm (ii) an on-line local search procedure (triggered by network congestion detection) based on the deletion of the lightest loaded lightpath and used for improving the resource utilization to allow the accommodation of incoming LSP requests. Compared to the previous two schemes, no details are included here to describe when a link can be considered congested.

3.1.3 Local Search approach

VTR schemes discussed in this section are based on Local Search techniques, a family of heuristics aimed at finding near-optimal solutions to hard problems, optimizing the value of the cost function by local modifications of the system configuration. A set of neighboring configurations is explored via a single application of some simple reconfiguration operation (a *local move*).

Labourdette et al. [66] propose the application of branch-exchange operations for a Local Search heuristic which move the current virtual topology to the optimal logical configuration in small steps in order to reduce the impact on the network during reconfiguration. The main drawback of this approach is that it gives rise to a multistep reconfiguration process that can take a long time and leads to topologies that are obsolete by the end of the process, especially with highly dynamic traffic patterns. A further limitation is that intermediate reconfiguration steps may result in a temporarily disconnected network. Even though based on Local Search techniques, this proposal can be considered part of the so-called “direct” approaches.

Narula-Tam et al. [81] propose an iterative reconfiguration algorithms which guarantees dynamic load balancing in ring networks. At each reconfiguration step, only a small change to the virtual topology is performed by using branch-exchange operations, hence minimizing the network disruption. An advantage of 3-branch exchange is that it never disconnects a ring network. The reconfiguration of the network is performed at regular intervals in order to track rapid changes in traffic patterns. Even if the authors prove that the proposed algorithm achieves performance improvements very close to optimal in the case of dynamic traffic, the main limitations of this scheme is that it is valid only for ring topologies. Another limitation of this approach is that the probability of network disruption is not completely neglected since the process of reconfiguration depends on the availability of wavelength resources on affected routes.

3.1.4 A comparison

The main advantage of direct reconfiguration approaches is that they lead to virtual topology configuration which are very close to optimal, but they are generally based on computationally costly heuristics because the underlying optimization problems are NP-hard. Even if both sacrifice the optimality of the solution, partial reconfiguration and Local Search approaches allow instead the implementation of heuristics with faster convergence.

Another important parameter to compare these approaches is the degree of network disruption incurred. Direct approaches are characterized by the highest impact on traffic disruption since they completely ignore the current virtual topology. Local Search approaches instead guarantee minimal network disruption since they are based on very simple reconfiguration operations. The impact of partial reconfiguration approaches depends by the choice of specific parameters such as: number of lightpaths added or removed, reconfiguration time interval, load threshold triggering the reconfiguration process, etc...

In Chapter 4 a novel reconfiguration algorithm based on Local Search is proposed and its performance in dynamic traffic condition is evaluated on irregular mesh topologies.

3.2 Traffic Engineering in MPLS networks

Most of the proposed TE schemes in MPLS networks are *preventive* according to the previous classification, they allocate paths in the network in order to prevent congestion, while only few are *reactive* which means they act only when problems start to appear. The two best known preventive mechanisms in the literature are Constraint-Based Routing (CBR) and traffic splitting. The first has its roots in the well-known Quality-of-Service routing problems in IP networks and refers to the calculation of LSP paths subject to various constraints (e.g. available bandwidth, maximum delay, administrative policies). The sec-

ond mechanism, traffic splitting, balances the network load through optimal partitioning of traffic to parallel LSPs between pairs of ingress and egress nodes. The preventive behavior of these proposals leads to a common drawback: when LSPs are set-up and torn-down dynamically, these schemes can lead to inefficiently routed paths and to future blocking conditions over specific routes. An overview of these two mechanisms is given in Sect. 3.2.1 and 3.2.2 respectively. Reactive congestion control schemes are usually based on mechanisms which allow the rerouting of existing LSPs towards less congested network links, and an overview of them is given in Sect. 3.2.3.

3.2.1 Constraint-based routing schemes

One of the better performing CBR schemes, called MIRA (Minimum Interference Routing Algorithm), is based on an heuristic dynamic online path selection algorithm [60]. The key idea, but also the intrinsic limitation of the algorithm, is to exploit the *a priori* knowledge of ingress-egress pairs to avoid routing over links that could "interfere" with potential future paths set-up. These "critical" links are identified by MIRA as links that, if heavily loaded, would make it impossible to satisfy future demands between some ingress-egress pairs. The drawbacks are: (i) the identification of the "critical" links leads to a severe computation complexity caused by the maximum flow calculation performed each time a new LSP has to be established (ii) the algorithm cannot estimate bottlenecks on links that are "critical" for clusters of nodes [112] (iii) MIRA can lead to an unbalanced network utilization because it does not take into account the current traffic load in routing decisions [23].

Many other proposals have used MIRA algorithm as a reference for comparison, mainly focusing their efforts on reducing the computation complexity of MIRA by using smarter mechanism to identify "critical" links in a network where the ingress-egress pairs are known in advance. One of the most interesting algorithm is DORA [20]. This scheme presents a reduced computation

complexity thanks to a smart mechanism to identify “critical” links in a network where the ingress-egress pairs are known in advance. The idea is to associate a path potential value (PPV) array to each ingress-egress pair, and using this information with the residual link bandwidth to compute a weight-optimized network path. PPV values are assigned according to a simple observation: the data between a given ingress-egress pair could flow through many different paths, and some links are more likely to be included in a path than others. PPV values indicates the potential of a link to be included in a path than other links. The link weight is obtained combining PPV values and current residual bandwidth of each link. This scheme allows furthermore to balance the load in the network, reducing the congestion probability.

Iliadis et al. propose a class of “Simple MIRA” algorithms having performance comparable with MIRA but with a reduced overall computational complexity [55]. Wang et al. proposal overcomes some of MIRA’s shortcomings by taking into account the bandwidth blocking effects of routing an LSP request [112].

All schemes cited so far considered only the setting up and routing of bandwidth guaranteed LSPs (tunnels). This means that QoS constraints such as delay and losses incorporated in SLAs are supposed to be converted into an effective bandwidth requirement for the LSP [60]. On the other hand, with the increase of real-time applications in the Internet, providing bandwidth guarantees only is not sufficient, but it is important to provision at least delay guarantees as well. Banerjee et al. [16] provide two classes of CBR schemes, one considering only the setup of bandwidth guaranteed LSPs as MIRA does, the other considering delay and bandwidth guaranteed LSPs. Lim et al. [69] propose a QoS routing scheme that gives priority to multimedia traffic with a reduced frequency of QoS state exchanges in the network compared to MIRA and other well-known algorithms. Three different load balancing algorithm for MPLS TE are proposed by Long et al. [71], whose key idea is to reduce congestion probability, mapping

the traffic with lower bandwidth demand over underutilized routes and leaving the shortest paths to LSPs with higher bandwidth requirements. Moh et al. [77] propose a new TE scheme based on a constraint-based routing algorithm that route traffic on a per-class level for setting up LSPs with delay constraints and bandwidth guarantees. The key idea is to map traffic trunks associated with the highest service class to the best LSPs, thereby guaranteeing their QoS requirements.

3.2.2 Load balancing schemes based on traffic splitting

Another important approach to TE and congestion control in MPLS networks deals with the optimal assignment of traffic to parallel LSPs between pairs of ingress and egress routers. A stochastic framework for the traffic splitting problem in MPLS networks is proposed in [38], while MPLS Adaptive Traffic Engineering (MATE) [44] focuses on engineering traffic across multiple explicit routes for a pair of ingress and egress routers. Although the main objective of traffic splitting is to balance the load in the network, it is not always possible for a routing algorithm to split traffic in an arbitrary manner since the traffic could be intrinsically unsplittable [60]. With this practice, packets from certain traffic flow can go through different paths experiencing variable delays thereby leading to packet reordering problems.

In the rest of the thesis we are not going to consider traffic splitting methods for TE in MPLS networks.

3.2.3 Load balancing schemes based on LSP rerouting

Only a few *reactive* congestion control schemes have been proposed in literature. It is interesting to notice that most of these schemes are analogous to VTR algorithms for optical networks, where LSPs are rerouted instead of lightpaths. Holness et al. [54] propose a mechanism called Fast Acting Traffic Engineer-

ing (FATE): the ingress LER and the core LSR react on information received from the network regarding flows experiencing significant packet losses by dynamically routing traffic away from a congested LSR to the downstream or upstream underutilized LSRs. The authors describe the procedure for congestion detection and its impact on the signalling mechanisms, but do not include any simulation on network performance. A method that considers elastic traffic and exhaustive search is proposed by Casetti et al. [25]. Jüttner et al. [58] propose an algorithm for the optimal routing of new LSPs based on the re-routing of an already established LSP when there is no other way to route the new one: at higher network utilization levels, on-demand CBR-based LSP setup can experience failures. In order to fit the new LSP demands, instead of a global reoptimization of all LSP paths, one quickly reoptimizes a single LSP. The algorithm is based on an Integer Linear Programming (ILP) formulation of the rerouting problem and an heuristic method for practical cases is proposed because of the excessive computation required by ILP. The simulations consider only static paths that stay in the network forever once established. Unfortunately the authors do not specify the traffic model used to run the algorithm, thus complicating comparisons with similar schemes.

Two novel schemes are proposed in Chapter 5 to reduce the congestion in an MPLS network by using load balancing mechanisms based on different local search heuristics.

3.3 Traffic grooming

The tremendous advancement of high-speed transmission technology, which allows data rates from 2.5/10 up to 40 Gbps, creates a large gap between the capacity of an optical channel and the bandwidth requirements of a typical connection request, which can vary in range from tens or hundreds of Mbps (e.g. STS-1 or Fast Ethernet) up to the full-wavelength capacity (e.g. 10 Gigabit

Ethernet). Furthermore the amount of wavelength channels available for most of the networks of practical size is much lower than the number of source-destination connections that need to be made.

Traffic Grooming (TG) poses the problem of how to multiplex (and demultiplex) a set of low-speed traffic streams onto high-capacity channels and switching them at intermediate cross-connects. Therefore the main objective of TG is to improve the wavelength utilization in the network and to minimize the network cost. Different multiplexing techniques can be used for TG in different domains of a wavelength-routed network [128]:

- Space-Division Multiplexing (SDM), which divides the physical space to increase the available bandwidth (e.g. a set of fibers into an optical cable);
- Frequency-Division Multiplexing (FDM), which divides the available frequency spectrum into a set of different channels. FDM in optical networks is known as WDM, which partitions the available wavelength spectrum in coarse channels, known as *waveband*, which are further divided into thinner wavelength channels;
- Time-Division Multiplexing (TDM), which divides the bandwidth time domain into different time-slots of regular size. A single wavelength channel can therefore be shared by different signals if they are not time-overlapped, i.e. as in SONET/SDH.
- Dynamic Statistical Multiplexing (DSM), which shares a single wavelength channel between multiple IP elastic traffic flows.

Even though most of the literature on TG problem implicitly consider how to groom low-speed connections onto optical channels using a TDM approach, the problem can consider whatever variation based on the multiplexing techniques described above. For example in [31], Cinkler discusses the advantages and drawbacks of TG while using a WDM approach (known as λ-grooming).

Chapter 7 describes for the first time to our knowledge the impact of a dynamic statistical multiplexing approach on TG.

Traffic Grooming can be decomposed in different problems, according to the nature of the traffic demand (static or dynamic) and to the physical topology under study. Most research on TG have concentrated mainly on SONET/WDM ring networks, both in static and dynamic traffic conditions, and the reader is referred to [42] for a good overview. The objective function is to minimize the total network cost, usually measured in terms of the number of Add-Drop SONET/SDH multiplexers.

However, as core networks migrate from rings to meshes, TG on optical networks based on arbitrary topologies has started to get more attention[1]. As for ring networks, traffic can be static or dynamic, therefore different kind of problems can be envisioned: network planning and topology design, for the static TG problem, and dynamic circuit provisioning for the dynamic TG problem. References [33, 127, 126, 36] reported some work on static TG for mesh topologies, which are not of interest in this thesis framework. Dynamic traffic grooming can be considered a *preventive* Traffic Engineering scheme, since it is a routing problem in a multi-layer network whose objective is to minimize the network resources used for each request, which implicitly attempts to minimize the overall blocking probability.

While in theory the constraint-based routing schemes for MPLS TE could be adapted to optical networks, in practice the key distinctions between pure IP/MPLS networks and G-MPLS based optical networks suggest us to be careful when CBR has to be applied in this last context. The fixed granularity of the wavelength labels, their discrete bandwidth and the technological limitations of wavelength converters which reduce their applicability in the optical networking context are only some of the major constraints that invalid most of

[1]Recently, some research on TG in regular topologies such as paths, stars and trees has been performed as well, whose main objective is to find theoretical bounds [40]

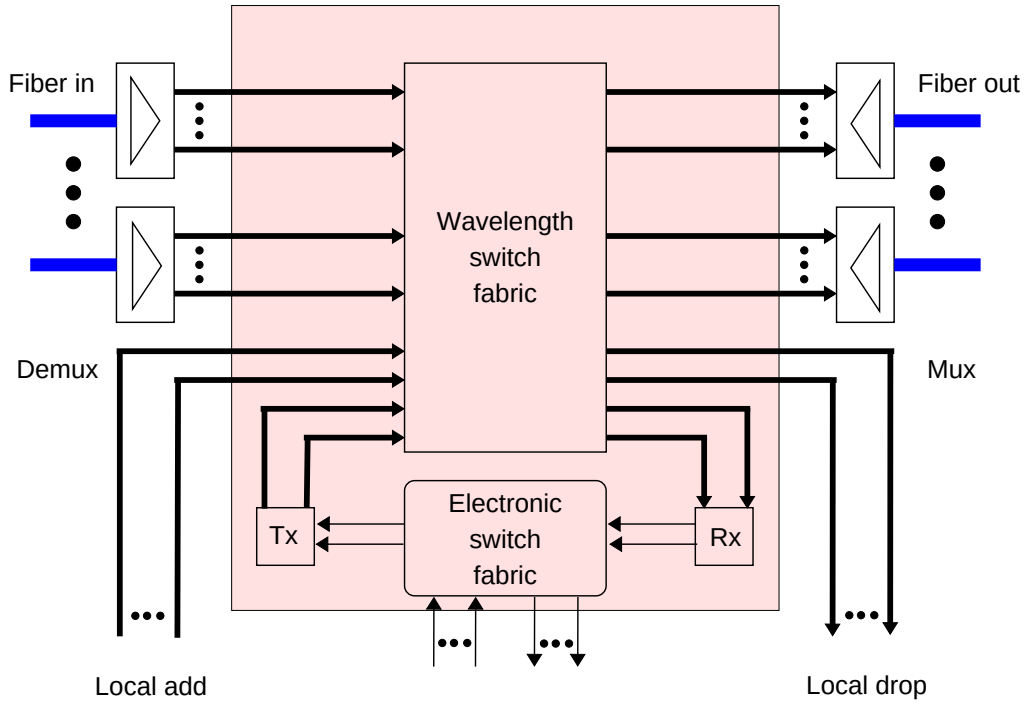


Figure 3.1: A multi-hop partial grooming OXC architecture

the schemes described in Sect. 3.2.1.

In the rest of the section an IP over Optical network architecture such as the one described in Sect. 2.3 is considered. In such networks, optical cross-connects are known to be the most important network elements, since they can establish or release connection requests in a matter of seconds (or even milliseconds). Crossconnects can be divided in transparent OXCs, if they perform all-optical (O-O-O) switching, or in opaque OXCs, if they perform switching through optical-electronic-optical (O-E-O) conversion. OXCs can be based on different architectures and technologies which result in different multiplexing and switching capabilities. An optical crossconnect can therefore have diverse capabilities for grooming low-speed traffic flows onto high-capacity wavelength channels, leading to four possible OXC's categories [129]:

- **Non-Grooming OXC**. This cross-connect has wavelength switching capability but it is not equipped with low data-rate ports, therefore no grooming

based on TDM or DSM approaches is possible.

- **Single-hop Grooming OXC**. This cross-connect has wavelength switching capability but it is equipped with some low data-rate ports. It can therefore multiplex (or demultiplex) low-speed traffic flows onto wavelengths but it cannot switch them when used at intermediate nodes.

- **Multi-hop partial Grooming OXC**. As shown in Fig. 3.1 the switch fabric of this OXC is composed of two parts: a wavelength switch fabric (either transparent or opaque) and an electronic switch fabric that can switch low-speed traffic flows (it can be either an IP or MPLS router or a SONET/SDH digital crossconnect). This type of OXC can switch traffic flows from different wavelength channels as well as groom them with other flows. In this architecture only few wavelength channels passing through the crossconnect can be switched to the electronic fabric for switching at finer granularity. The number of ports connecting the electronic fabric to the wavelength fabric determines the degree of grooming capability of the OXC.

- **Multi-hop full Grooming OXC**. This cross-connect provides full grooming functionality since each wavelength crossing it is fully demultiplexed and all the low-speed traffic flows are switched by the electronic fabric.

The interconnection of one or more types of OXCs may create different grooming scenarios for a core optical network. When the crossconnects considered are all Non-Grooming OXCs, the problem is reduced to the classical Routing and Wavelength Assignment, where each connection request corresponds to an entire lightpath. When studying the impact of some grooming algorithm, there could be networks with sparse or full grooming capabilities with both partial or full Grooming OXCs (G-OXCs). In this case, an incoming sub-wavelength connection request could be routed over a direct lightpath (a *single-hop* path at the IP level) connecting an ingress router to an egress router

or over a sequence of lightpaths (a *multi-hop* path at the IP level), crossing many intermediate G-OXCs along its route.

Dynamic traffic grooming schemes can be distinguished according to the interconnection model considered in the IPO network under consideration. In particular both overlay and integrated TE approaches are possible, according to the amount of information exchanged between the IP and the optical layer. Next two sections give an overview of the most important contributions for both TE approaches.

3.3.1 Overlay dynamic grooming

Overlay dynamic grooming refers to dynamic provisioning schemes which guarantee the routing of incoming traffic request into an IPO based on the *overlay* interconnection model. As described in Sect. 2.3.1, each layer runs its own routing instance; therefore an IP router triggers the set-up of new lightpaths through some User-Network Interface, but it would route the incoming traffic flow requests over the current virtual topology by using some well-established Internet-based routing protocol such as OSPF or IS-IS (or their TE extensions if the routers have MPLS forwarding engines). Similarly, each time a new lightpath needs to be set-up, the optical layer would find a path for the optical channel by using some centralized or distributed dynamic RWA algorithm (see Sect. 2.2.2).

The main advantage of such schemes is that they are very practical in the short-term, since UNI interfaces have been already standardized and tested in laboratories, as well as effective RWA algorithms to dynamically find paths in a mesh optical network. Clearly, due to the reduced coupling between the two layers, network resource usage could not be optimized. Better said, the responsible for each layer (typically an Internet Service Provider for the IP layer and an Optical Backbone Provider for the WDM layer [94]) would try to maximize the usage of its own network resources, therefore a global network optimization

would never be possible.

Even if the peer and augmented models do not seem realizable in the near future, most of the dynamic grooming algorithms proposed in the literature implicitly consider such models, while only a few based on the overlay model have been proposed so far. In the following, two overlay schemes based on a TDM approach to traffic grooming are illustrated. Therefore it is always assumed that all IP routers generating traffic requests are equipped with MPLS functionalities to generate bandwidth-guaranteed LSP connections.

Assi et al. [10] propose a dynamic grooming scheme which is based on constraint-based routing algorithms in the IP layer. Constraint routing is further augmented by dynamically routing both an active and another alternate link/node disjoint back-up path in order to provision a given connection requests. While keeping a fixed dynamic RWA algorithm in the optical layer, the performance of three different CBR algorithms while varying the number of wavelengths per fiber link is evaluated in term of blocking probabilities. This scheme is valid only for IPO networks with full grooming capabilities and based on multi-hop full grooming OXCs.

Niu et al. [82] instead investigate the effect of the number of add/drop ports of OXCs on the LSP blocking performance, by examining three connection admission control (CAC) policies for the establishment of LSPs. A lower bound for the number of add/drop ports to achieve the best blocking performance is shown by simulation on different network topologies. MinHop routing algorithm is considered in the IP layer, and a straightforward SPF (shortest-path-first) in the optical layer. It is furthermore studied the impact of having OXCs with full- or no-wavelength conversion capability.

A common difficulty which characterize all the dynamic grooming schemes (overlay and integrated) in literature is a formal framework to describe the routing algorithm themselves.

In Chapter 7 a formal framework for the definition of dynamic grooming

policies in IPO networks is given for the first time to our knowledge and then specialized for the overlay interconnection model. A family of overlay dynamic grooming policies is defined and their performance analyzed by using realistic traffic models according to a Dynamic Statistical Multiplexing approach to TG.

3.3.2 Integrated dynamic grooming

Integrated dynamic grooming refers to dynamic provisioning schemes which guarantees the routing of incoming traffic request into an IPO based on the *peer* (or *augmented*) interconnection model. The IP and optical networks are treated together as a single integrated network, where IP routers and OXCs are considered peer network elements, thus the topology and other network information are completely (or partially) shared in a unified control plane. As described in Sect. 2.3.3, the G-MPLS framework and all its own routing and signalling protocols could match such requirements. In fact, the usage of link-state routing protocols such as OSPF-TE allows the construction of *auxiliary graphs* representing the network status (e.g., in term of wavelength occupancy and usage, or other information such as the number of add/drop ports used per OXCs,...), over which some Dijkstra-based algorithm could be applied to find a path in the network.

The main advantage of these schemes is that they guarantee an optimized usage of network resources since the design and operation of the optical layer and the IP layers are integrated; the drawbacks are those already listed for the peer and augmented interconnection model regarding the security issues related to the amount of routing information passed from one layer to the other. Another potential problem rises when considering errors due to inaccurate routing information [74]; due to the great amount of information to be exchanged in a G-MPLS network, this kind of errors could dramatically compromise network performance.

Even though these schemes can be considered practical in the long term only,

most of the research has been focusing on them, in order to investigate schemes which can maximize the network usage. All of the algorithms presented in the following are based on a TDM approach to traffic grooming. Therefore it is always supposed that all IP routers generating traffic requests are equipped with MPLS functionalities to generate bandwidth-guaranteed LSP connections.

Kodialam et al. [63] propose one of the first integrated grooming algorithms, known as Maximum Open Capacity Allocation (MOCA), which extends some of the ideas behind MIRA algorithm (see Sect. 3.2.1) to IP over Optical networks. By considering an auxiliary graph which takes into account both IP and WDM layer information, MOCA algorithm determines whether to route an arriving LSP request over the existing topology or whether is better to open a new lightpath. Then, for routing over the existing IP layer topology, it computes "good" paths, while for new wavelengths setup it computes "good" lightpaths. The "goodness" of a path is based on extensions of the MIRA algorithm: in order to maximize the future requests allocation, it is better to route new LSPs along paths which minimize the "interference" or maximizes the open capacity between an ingress-egress routers pair. Unfortunately, due to the close behaviour of MOCA to MIRA, this algorithm suffers from the same main drawbacks already identified in Sect. 3.2.1.

A similar auxiliary graph is described in [106], where Srinivasan et al. evaluate the performance of well-known routing algorithms such as shortest-widest path, widest-shortest path and available shortest path in IPO networks with both sparse or full grooming capability. The crossconnects considered in the network are all multi-hop full grooming OXCs. The authors describe the impact of different traffic granularities on the network performance, and observe that increasing the grooming capability in a network could result in degrading the performance of the WSP routing algorithm.

The main limit of both these methods is that they do not consider the impact of an OXC architecture characterized by partial grooming capabilities, which is

the most practical case for cost reasons. In [125] Zhu et al. extend the graph model proposed in [126], which allows the description of IPO networks based on different OXC node architectures, to develop a novel grooming policy called Adaptive Grooming Policy (AGP). By properly adjusting the weights of the edges in the proposed auxiliary graph, it is possible to propose different grooming policies, characterized by different objectives such as minimizing the number of traffic hops in the virtual topology or in the physical topology, or even minimizing the number of used lightpaths.

While all the previous contributions are valid for the peer interconnection model only, Koo et al. [64] propose a novel dynamic grooming algorithm, called CAPA_AUG, which can be applied for IPO networks based on an augmented model. The main advantage of this proposal is that, instead of considering full information sharing between the two layers, the authors propose an auxiliary graph whose edges are weighted with some aggregated information regarding capacity information from the WDM layer. Furthermore, the proposed model takes into consideration the hypothesis of having multi-hop partial grooming OXCs. It is also shown that CAPA_AUG outperforms MOCA [63] by an order of magnitude in term of blocking probability. In the same paper a dynamic grooming algorithm (named MLH_OVLY) for the overlay model is proposed as well, however it must be noted that this algorithm requires the exploration of the optical connectivity between all the possible source-destination node pairs in the network to decide whether to accept or block a new connection, which seems hardly feasible in large networks, at least if no information from the optical level is available at the IP level as in the augmented or peer model.

Mohan et al. [111] propose a dynamic grooming algorithm for the peer model who considers routing of prioritized LSPs by providing service differentiation between classes of high and normal priority traffic. The authors propose an auxiliary graphs to represent network status and cost of ports usage. A CAC-based routing algorithm which admits high-priority LSPs in preference over

normal-priority LSPs is developed and its effectiveness proved by simulations. The main disadvantage of this admission-control technique is how to decide the right amount of resources dedicated to normal-priority traffic, which can lead to block high-priority requests even if the network could support them.

In Chapter 6 a novel integrated TE scheme is described whose main objective is to route connection requests with QoS constraints. Delay and packet-loss ratio requirements are translated into constraints at the physical layer and an efficient service differentiation is provided through a sub-optimal connection preemption algorithm.

Chapter 4

Dynamic Load Balancing in WDM Networks

Load Balancing in WDM based networks is a TE mechanism to reduce the congestion in the network. In such networks, reducing congestion implies that a certain number of spare wavelengths are available on every link to accommodate future connection requests or to maintain the capability to react to faults in restoration schemes. In addition, reducing congestion means reducing the maximum traffic load on the electronic routers connected to the fibers.

Load balancing in WDM networks consists of two subproblems: the lightpath connectivity and the traffic routing problem. The routing problem has its origin at the beginning of the networking research, see [81] for a review of previous approaches to the problem. In particular, adaptive routing, that incorporates network state information into the routing decision is considered in [78] in the context of all-optical networks, while previous work on state-dependent routing with trunk reservation in traditional telecommunications networks is considered in [75]. It is also known that flow deviation methods [48], although computationally demanding, can be used to find the optimal routing that minimizes the maximum link load for a given network topology.

Because global changes of the logical topology and/or routing scheme can be disruptive to the network, algorithms that are based on a sequence of small steps

(i.e., on local search from a given configuration) are considered. In [65] "branch exchange" sequences are considered in order to reach an optimal logical configuration in small steps, upper and lower bounds for minimum congestion routing are studied in [118] that also proposes variable depth local search and simulated annealing strategies. Strategies based on small changes at regular intervals are proposed in [81].

In this chapter an investigation on protocols that consider IP-like routing strategies is described, where the next hop at a given node is decided only by the destination of the communication. In particular, we consider a basic change in the network that affects a single entry in a single routing table. In the context of all-optical networks this is relevant for optical packet switching networks, or for circuit switching networks (e.g. based on G-MPLS) where the optical cross-connects allow arbitrary wavelength conversion.

Let us define the context and the notation. By *physical topology* we mean the actual network composed of passive or configurable optical nodes and their fiber connections. The *logical topology* is given by the lightpaths between the electronic routers, determined by the configuration of the OADMs and transmitters and receivers on each node. The *traffic pattern* is available as an $N \times N$ matrix (N being the number of nodes in the network) $T = (t_{ij})$ where t_{ij} denotes the number of lightpaths (or the number of traffic load units) required from node i to node j. We assume that the entries t_{ij} are non-negative integers and $t_{ij} = 0$ if $i = j$. A *routing table* is an array, associated to each node of the network, containing the next-hop information required for routing. In the following we shall consider IP-like routing, where the next hop is maintained for each possible destination, regardless of the index of the source node. For a given traffic pattern and routing tables associated to the nodes, the sum of the number of lightpaths passing through each link is called the *load* of the link. Finally, the maximum load on each link of a path is called the *congestion* of this path. The maximum load on each link of a network is called the *congestion* of

the network.

The Load Balancing problem can be defined as follows.

> LOAD BALANCING — Given a physical network with the link costs and the traffic requirements between every pair of source-destination (number of lightpaths required), find a routing of the lightpaths for the network with the least congestion.

4.1 Local Search for the Load Balancing Problem

The basic idea of the new Load Balancing scheme is as follows: start from the shortest path routing and then try to minimize the congestion of the network by performing a sequence of local modifications. For each tentative move, the most congested link is located, and part of its load is re-routed along an alternate path.

We shall begin with some definitions and explanations of the functions and variables. We maintain the set *candidatePathSet* containing paths that are candidate to replace those passing through the most congested links of the network. This set is emptied at each iteration of the algorithm.

Given all routing tables, every node *d* identifies a spanning tree of the network, namely the tree composed of all links that carry lightpaths addressed to *d* (it is a tree because of the destination based routing). We are interested in identifying a subtree of this tree, and we use the function *routingTree*(*d*,*r*) returning the subtree rooted in node *r* of the routing tree having destination node *d*. The shaded tree shown in Figure 4.2 is actually *routingTree*(*dest*,*cFrom*), and it contains all nodes whose lightpaths directed to destination *dest* pass through node *cFrom*. The function *shortestPathRouting*(*network*) calculates the shortest path tree for each destination node and returns the corresponding set of routing tables as a matrix. *rTable*[*n*] is the routing table of node *n*, whose *i*-th entry *rTable*[*n*][*i*] is the next-hop node index for lightpaths passing through node *n*

```
1. rTable ← shortestPathRouting(network)
2. <congestion,congestedLinkSet> ← calculateLoad(network,traffic,rTable)
3. repeat
4.     bestCandidateLoad ← +∞
5.     candidatePathSet ← ∅
6.     for each link <cFrom,cTo> ∈ congestedLinkSet
7.         for each destination node dest such that rTable[cFrom][dest]=cTo
8.             for each source node src ∈ routingTree(dest,cFrom)
9.                 removePartialLoad (src,dest)
10.                for each neighbor node nb ∈ neighborhood(src)
11.                    vl ← load on the candidate path from src to dest through nb
12.                    if (vl = bestCandidateLoad)
13.                        candidatePathSet ← candidatePathSet ∪ {<src,dest,nb>}
14.                    else if (vl < bestCandidateLoad)
15.                        bestCandidateLoad ← vl
16.                        candidatePathSet ← {<src,dest,nb>}
17.                restorePartialLoad (src, dest)
18.    if ( candidatePathSet ≠ ∅ )
19.        <src,dest,nb> ← pickRandomElement (candidatePathSet)
20.        rTable[src][dest] ← nb
21.        <congestion,congestedLinkSet> ← calculateLoad(network,traffic,rTable)
22.    else exit
23. until MAXITER iterations have been performed
```

Figure 4.1: The Local Search RSNE algorithm.

and with destination *i*.

Finally, function *calculateLoad*(*network*,*traffic*,*rTable*) returns the network congestion given the network topology, the traffic pattern and the current routing scheme. The function also returns the set of links having maximum loads.

Figure 4.1 shows a description of our Local Search algorithm used for the Load Balancing problem. In the rest of the chapter we shall refer to it as *Reverse Subtree Neighborhood Exploration* (RSNE).

The initialization section (lines 1–2) starts by generating the routing tables by the application of the Shortest Path Routing algorithm to the specific network (the costs of all the edges are considered as uniform). Using the function *calculateLoad* we initially calculate the load on each link of the network, the initial value of *congestion* (from which the local search algorithm starts its re-

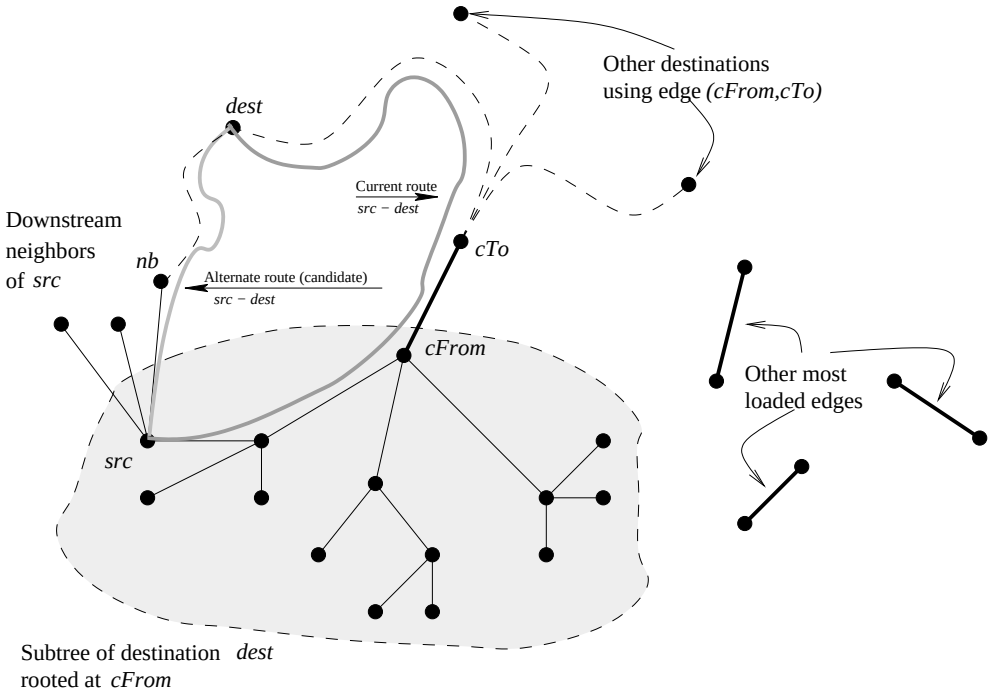


Figure 4.2: Search space for a move of the RSNE algorithm.

search of the minimum) and the set of congested links *congestedLinkSet*. Then *candidatePathSet* is empty at the beginning of each iteration.

Every iteration of the local search algorithm (lines 3–23) consists of two distinct parts. First, a set of alternate paths for part of the traffic passing through the most congested links is found (lines 6–17): then, once the most promising candidate move is selected, the routing tables of the network (lines 18–22) are corrected accordingly, then the iteration starts by considering the new congested links.

The first part includes the core of RSNE algorithm. Refer to Figure 4.2 for a visual reference. We consider each congested link in *congestedLinkSet* (loop at lines 6–17). Let us identify this link with its endpoints (*cFrom,cTo*). Then the procedure iterates through all lightpaths that use that link. Two nested loops are present: the first (line 7) scans the routing table of node *cFrom* looking for all destination nodes *dest* using that link. The second (line 8) scans all nodes *src*

whose lightpaths directed to *dest* run through *cFrom*. These nodes identify the subtree rooted in *cFrom* of the routing tree having destination *dest*.

For every (*src*,*dest*) pair whose lightpath goes through the link (*cFrom*,*cTo*), we try to reroute such lightpath by altering the routing table in *src*. To do this, we temporarily remove the load of the lightpath from the current route (function *removePartialLoad* at line 9) and iterate through all downstream neighbors *nb* of *src* calculating the maximum load that would be caused by re-routing the lightpath, provided that the new route does not end up in a cycle and that the congested edge is avoided. The best alternate paths, in terms of maximum load, are collected into the candidate set *candidatePathSet*. In particular, the current minimum is stored in *bestCandidateLoad*. If the load obtained after this traffic re-route is equal to *bestCandidateLoad*, then the re-route is added to the candidate set (lines 12–13), if it is smaller, the candidate set is re-initialized to the current re-route and its load is stored as the new best (lines 14–16). At the end of the alternate paths research, the partial load associated to the path originating in *src* and terminating in *dest* is reallocated (line 17), in order to allow the search of new paths with different source nodes *src* (line 8).

In the second part of the RSNE algorithm, if the resulting set *candidatePathSet* is not empty then one random solution is selected from it (line 19), and the routing table of the network is updated (line 20). Finally, a new value of *congestion* and the relative set of most loaded links *congestedLinkSet* is calculated again in order to start a new search of alternate paths through the network.

Note that the local search algorithm continues looking for better values of *congestion* until the set of candidate re-routes *candidatePathSet* is empty (line 22), or until a given number of iterations has been performed (line 23).

4.2 Algorithm properties and modifications

As an estimate of the CPU time required by the algorithm, let us consider its computational complexity of an iteration (lines 4–22) in terms of *node visits*, i.e. operations on nodes of the network. In detail, a counter is incremented for each node considered in the candidate path at line 11. The proposed algorithm is composed of nested cycles. Let n be the number of nodes in the network; if d is the largest node degree, the number of edges is $O(nd)$. The number of iterations for the loop at line 6 is only bounded by the number of links in the network. Each of the two nested loops at lines 7 and 8 may scan almost all nodes in the worst case. Function `removePartialLoad` may need to operate on a long path (the diameter of the network is potentially equal to n). Loop at line 10 is executed d times at most, and the path check at line 11 is again bounded by the number of nodes. Then the complexity of one `RSNE` iteration is $O(n^4 d^2)$. So in the worst case, when d grows linearly[1] with n, the overall complexity of an iteration is $O(n^6)$.

However, this is a pessimistic, worst-case bound: experiments on networks modeled by Euler disk graphs (see Sect. 4.4.2) suggest that the number of node visits (while the algorithm is exploring the routing subtree and checking the new path) has $O(n^\alpha)$ empirical complexity with α slightly higher than 2. However, simplifications of the algorithm shall be introduced in the next subsections in order to lower the worst-case complexity of the algorithm to a reasonable power of n and d.

4.2.1 Randomized version (`fRSNE`)

Computational complexity and scalability are two major issues in current network algorithm research: algorithms are required to adapt to different network

[1] this happens in many real-world contexts: in Internet autonomous systems there is evidence [110] of a strong correlation between the logarithms of n and d with a coefficient near to 0.95, so that $d = O(n^{.95})$.

sizes without excessive slowdown. In the present case, a drastic reduction of the computational complexity can be obtained by reducing some of the loops to a fixed, small number of random choices. A fixed-size subset of the congested links can be explored by the loop at line 6, and a fixed number of random destinations can be covered by the loop at line 7. The loop at line 8 is basically a subtree exploration. By limiting the degree of the descent through this tree, the number of source nodes that are explored can be reduced. When all these reductions are performed together, the complexity of an iteration becomes $O(n^2 d)$ in the worst case ($O(n^3)$ for unbounded degree). Experiments on the same Euler disk graphs have reported an $O(n^\beta)$ empirical complexity (in terms of node visits), with $\beta \approx 1.67$.

In the following, the randomized version will be called `fRSNE`(*e*,*d*,*s*) (*fast* `RSNE` on *e* edges, *d* destinations per edge and degree *s* of source subtree exploration). As shown in Sect. 4.4, when compared to `RSNE`, performance degradation of `fRSNE`(1,1,1) (the randomized version where one edge, one destination and just one source subtree path are considered) is almost negligible in terms of congestion, while average hop length and average edge load are only slightly increased.

4.2.2 Incremental version (`I-RSNE`)

Local search heuristics can be seen as stepwise refinements of an initial solution by slight modifications of the system configuration. In our case, the `RSNE` algorithm starts from a shortest path routing scheme and changes at every step a routing table entry of a single node in the matrix. By performing many such changes, the system stabilizes to a low congestion configuration.

This iterative scheme is appropriate for a dynamic environment where traffic requirements evolve with time. In particular, if changes in the traffic matrix are reasonably smooth[2] even a small number of steps of the `RSNE` algorithm in Fig-

[2]The assumption is reasonable even though IP traffic is known to be bursty: in fact, traffic requirements are

ure 4.1 are sufficient to keep the system in a suitable state as the traffic matrix changes. Of course, only lines 2–23 must be executed, because we don't want to restart from scratch by calculating the shortest path routing tables. Moreover, a very low number of iterations of the outer loop (lines 3–23) must be performed at each step, i.e. MAXITER must be very small to avoid excessive traffic disruption. In the following, we refer to the incremental algorithm as *Incremental* RSNE with *k* outer iterations per step: I-RSNE(*k*).

Simulations discussed in Sect. 4.4 show that even a single iteration of the algorithm yields good results under a fairly generic traffic model. The number of iterations of the algorithm is equivalent to the number of routing table entry modifications in the systems. Thus, a very limited number of routing table entries must be modified as traffic evolves in order to keep congestion at low levels. Moreover, I-RSNE is fully compatible with the fRSNE randomized scheme discussed above, and experiments show that the performance of the combination which shall be called I-fRSNE does not degrade.

A similar approach has been proposed in [104], where branch-exchange methods are proposed for a local search heuristic. However, the type of local modification is different from our proposal.

4.2.3 Restricted Neighborhood Exploration (RNE)

A simplified version of the algorithm was also tested, which can be called RNE (Restricted Neighborhood Exploration); it only considers node *cFrom* as a candidate for re-routing, with no exploration of its *routingTree* (consider the algorithm in Figure 4.1 where the loop at lines 8–17 is executed only once with *src* equal to *cFrom*). This would be equivalent to remove as much load as possible from the congested link with a single routing table change. However, as simulations in Sect. 4.4 show, such policy does not compare well with RSNE and fRSNE, probably because too large amounts of load are moved at each step,

given as an average over a certain amount of time, with some marginal capacity left to accommodate traffic peaks.

while finer modifications are more appropriate.

4.3 ILP formulation

In order to compare the results of the proposed techniques with the actual optimum, the following Integer Linear Programming (ILP) formulation can be used.

Let us consider the n-node oriented graph $G = (V, E)$ where $V = \{1, \ldots, n\}$ and $E \subseteq V \times V$. For each couple of nodes $s, d \in V$, $s \neq d$, let us define the requested traffic as T_{sd}.

For each $i, j, s, d \in V$ such that $(i, j) \in E$ and $s \neq d$, the binary flow variable $F_{sd}^{ij} \in \{0, 1\}$ is introduced; its value is 1 if and only if data from source s to destination d is routed through edge (i, j). The usual set of flow conservation (solenoidal) constraints can be imposed (each family of constraints is identified by an indexed label on the right):

$$\sum_{\substack{i \\ (i,j)\in E}} F_{sd}^{ij} - \sum_{\substack{i \\ (j,i)\in E}} F_{sd}^{ji} = \begin{cases} -1 & s = j \\ 1 & d = j \\ 0 & \text{otherwise,} \end{cases} \qquad \text{(FLOW}_{sd}^{j}\text{)}$$

one equation for each combination of $j, s, d \in V$ with $s \neq d$. These constraints avoid imbalances between the incoming and outgoing flow at all nodes with the exception of the source s and the destination d.

IP routing is destination-driven, so a new family of binary variables is introduced in order to consider only the flow destination. For each $d, i, j \in V$ such that $(i, j) \in E$, let $R_d^{ij} \in \{0, 1\}$ equal to 1 if and only if edge (i, j) carries traffic towards destination d. This information can be extracted from the F_{sd}^{ij} variables by imposing $R_d^{ij} = 1$ as soon as there is at least one source s such that $F_{sd}^{ij} = 1$. This can be obtained by the following linear constraints:

$$R_d^{ij} \geq F_{sd}^{ij}, \qquad \text{(ROUTE}_{sd}^{ij}\text{)}$$

one equation for every combination of indices $s, d, i, j \in V$ where $s \neq d$ and $(i, j) \in E$.

IP routing is achieved when for every node and every destination there is at most one outgoing edge carrying flow for that destination, regardless from the source. The following constraints impose that limit:

$$\sum_{\substack{i \\ (i,j)\in E}} R_d^{ij} \leq 1, \qquad (\text{IP}_d^i)$$

one equation for every $i, d \in V$.

After imposing IP routing on the graph, congestion must be minimized by introducing a new variable F_{max}, constrained in order to be larger than every link load:

$$F_{\text{max}} \geq \sum_{\substack{s,d \\ s\neq d}} T_{sd} F_{sd}^{ij}, \qquad (\text{LOAD}^{ij})$$

for every $(i, j) \in E$.

The problem can be stated as follows:

$$\text{Minimize} \quad F_{\text{max}}$$

$$\text{Subject to} \begin{cases} \text{FLOW}_{sd}^j & \forall s, d, j \in V, \quad s \neq d \\ \text{ROUTE}_{sd}^{ij} & \forall s, d, i, j \in V, \quad s \neq d, \quad (i, j) \in E \\ \text{IP}_d^i & \forall d, i \in V \\ \text{LOAD}^{ij} & \forall i, j \in V, \quad (i, j) \in E. \end{cases} \qquad (\text{ILP})$$

This formulation will be used in the following section in order to test the RSNE heuristic against the optimal value on small networks, and against lower bounds determined by the CPLEX optimizer if the optimum search could not be completed because of time limits.

4.4 Simulation results

Several experiments have been performed in order to test the RSNE heuristic on different network sizes and topologies, both computer generated and real, with static and dynamic traffic conditions.

4.4.1 Experimental setting

Shortest path and the proposed heuristics were simulated by a C++ program, with some classes devoted to generate network and traffic instances according to various models. The following network models were considered:

- *random graphs*, parameterized by the number of nodes and edge density, i.e. the probability of an edge to exist for every couple of nodes;
- *Euler disk graphs*, parameterized by the number of nodes and by a radius r, where nodes are scattered in a unit square, and two nodes are connected if and only if their distance is less than r;
- *real-world networks*, with connection matrices read from a file.

Traffic models are of the following types:

- *static uniform matrices*, where all non-diagonal entries have the same value;
- *static random matrices*, where all non-diagonal entries are taken by a uniform random distribution between a given minimum and maximum;
- *dynamic random matrices*, generated as follows (see [81] for a similar model): given two positive integers N and Δ, consider a sequence of $N\Delta + 1$ traffic matrices $(T^0, T^1, \ldots, T^{N\Delta})$ where matrices $T^{k\Delta}$, $k = 0, 1, \ldots, N$ are random and independently generated, by choosing a random maximum value between 10 and 100 and calculating every entry as a

random number between 10 and this maximum; all other matrices are linear interpolations of the immediately adjacent random matrices; in other words, given $h = 0, \ldots, \Delta - 1$ and $k = 0, \ldots, N - 1$, entry $T_{ij}^{k\Delta+h}$ of matrix $T^{k\Delta+h}$ is computed as follows:

$$T_{ij}^{k\Delta+h} = \text{round}\left[\left(1 - \frac{h}{\Delta}\right) T_{ij}^{k\Delta} + \frac{h}{\Delta} T_{ij}^{(k+1)\Delta}\right];$$

- *real-world traffic matrices*, read from a file.

To generate pseudo-random sequences, a Mersenne Twister algorithm of period $2^{19937} - 1$ was used[3]. Every random-sensitive object (the graph generator, the traffic generator and the heuristic routing algorithms) was endowed with a different instance of the generator, in order to ensure independence and reproducibility of initial conditions for different algorithms.

The C++ program was also used to output problem instances to files in MPS format in order to solve them via a linear problem optimizer. CPLEX 7.1 was used to solve these instances.

4.4.2 Empirical complexity tests

The RSNE and fRSNE algorithms have been tested on random Euler disk graphs in order to obtain a measure of the growth of computational time (in terms of node visits per iteration) as the network size increases. Euler disk graphs were selected because they show some properties similar to real-world networks, such as local connection schemes and a larger number of multi-hop paths when compared with completely random graphs. Moreover, by letting the number of nodes n increase while keeping the radius constant, the degree is proportional to n, thus unbound.

[3]The code, written by Makoto Matsumoto (Keio University, Japan) and Takuji Nishimura (Yamagata University, Japan), is available at:
`http://www.math.keio.ac.jp/matumoto/emt.html`.

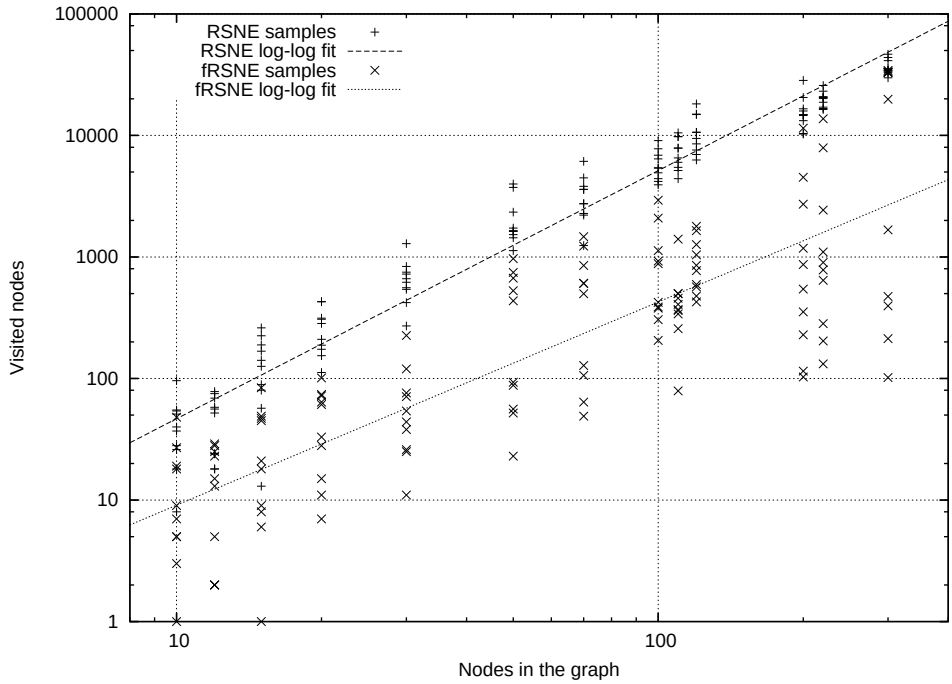


Figure 4.3: Number of node visits in one iteration of RSNE and fRSNE for Euler disk graphs with radius .3 on a unit square.

Figure 4.3 plots the number of node visits against the size (in nodes) of the network: 10 samples for each network size have been generated, and both RSNE and fRSNE have been tested. Least-squares linear regression has been calculated on the logarithmic transforms of the data to estimate the highest exponent in the dependence formula. If v is the number of node visits, the resulting dependencies are $v \approx 0.14 \cdot n^{2.04}$ for the RSNE algorithm and $v \approx 0.02 \cdot n^{1.67}$ for fRSNE. Thus, even though experimental data show a large variability, the fRSNE algorithm has a lower asymptotic complexity in the case considered as well as a much lower constant multiplier.

4.4.3 Tests on static traffic

The first tests of the RSNE algorithm aim at assessing its capacity to outperform the simple shortest path scheme in a static traffic context.

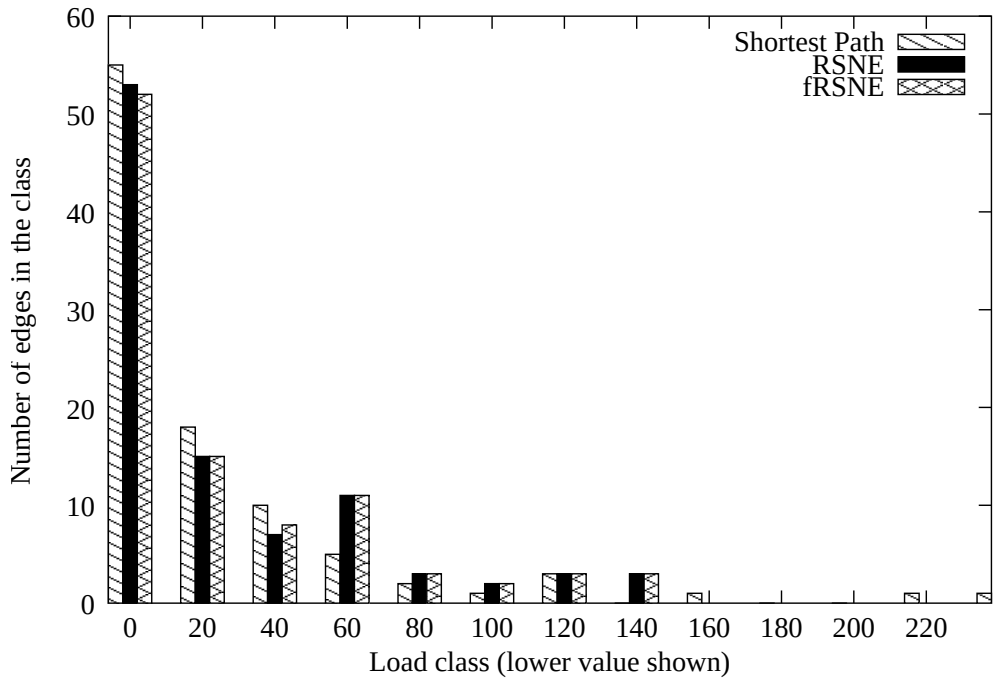


Figure 4.4: Distribution of edge loads for Shortest Path routing, RSNE and fRSNE.

The 24-node regional network presented in [124] and the traffic pattern presented in the same work have been chosen for the first static simulation. We executed 1000 steps of the RSNE and fRSNE schemes and compared the results with the initial shortest path configuration.

Figure 4.4 shows how edge loads distribute under the three policies for a random traffic matrix. Load values have been grouped in classes of 20, and the histogram represents the population of each class. The maximum load for the shortest path routing was 242, RSNE and fRSNE both reduced it to 157 (RSNE found it at the 26th iteration), obtaining a 36% reduction. The overall shape of the load distribution is not much different in the three cases, apart from the longer tail in the shortest path distribution. In particular, the number of light-loaded edges remains similar[4], and the average edge load is increased by 6.8%, from 30.37 to 32.42.

[4]Indeed, a scheme which reduces congestion but substantially increases the load of many other edges would be unacceptable.

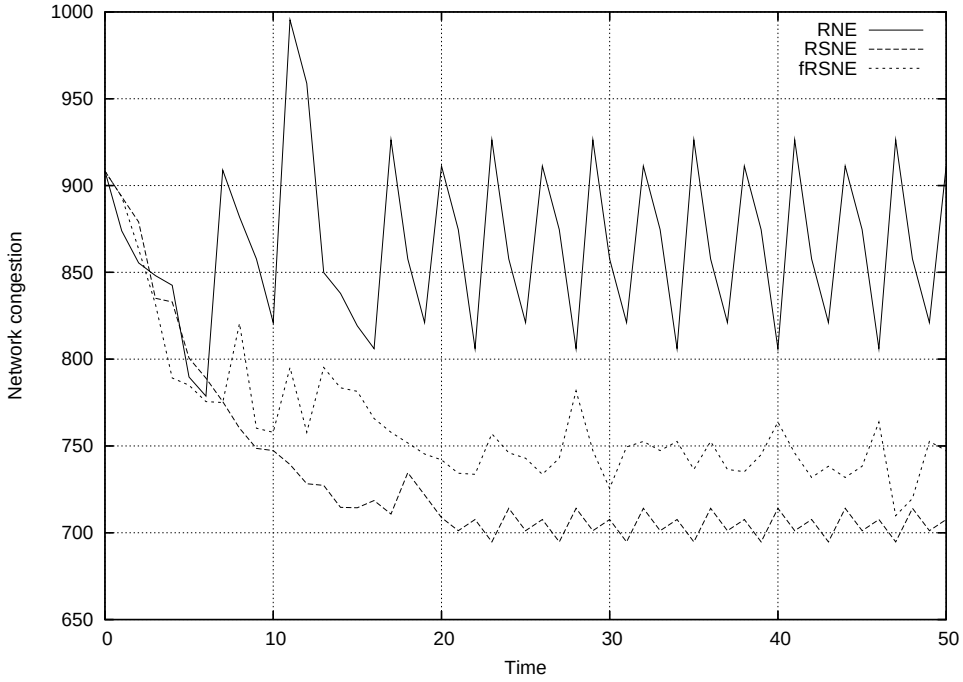


Figure 4.5: Behavior of heuristics during a single run.

Another important verification concerned the distribution of hop lengths. In this case, the maximum hop length of 7 hops was left unchanged by both RSNE and fRSNE algorithms, while a 1.8% increase in the average hop length (from 2.77 to 2.82 with RSNE) was verified. This is unavoidable, because the shortest path routing minimizes hop length by definition, so it necessarily outperforms all other schemes. However, the imbalance is very small.

Figure 4.5 plots the best congestion value against the number of steps for one run of the RNE and RSNE algorithms; here the NSFNET topology was used with a random traffic pattern from 10 to 100 for each couple of nodes. It turns out that the more complete RSNE algorithm outperforms its simplest RNE version, although the latter seems to have a better result in its earlier phase: this behavior shows up in many cases, probably because the algorithm is forced to move larger portions of load from edge to edge, achieving temporary better results but ending up with a complex, non-improvable routing scheme. For

Table 4.1: Comparison between algorithms on small random networks with 60% density. Average figures on 10 experiments per size, intervals are shown where ILP could not find solution in scheduled time.

Nodes	Shortest Path	RSNE	ILP
5	333.83	312.41	312.24
6	379.05	348.91	340.12
7	345.86	263.98	[257.16, 254.51]
8	416.03	325.24	[345.77, 305.92]
10	374.87	254.88	[374.87, 228.45]
12	463.10	249.99	[495.73, 249.13]

clarity, only the first 50 iterations are shown. However, the fRSNE scheme, showing an intermediate behavior due to the smallest move space at each step, eventually finds the same values shown by the complete heuristic. The above simulation gave a maximum hop length equal to 7 (i.e. a lightpath needs to travel 7 links from source to destination) at each iteration. This is the minimum, because it also results from the shortest path routing assignment that initiates all algorithms.

Table 4.1 shows the results of a comparison between the shortest path algorithm, the RSNE technique and the solutions provided by the CPLEX optimizer when the same instances were put in the ILP form. Every line reports the average of 10 experiments; a time limit of 10 minutes for each instance was imposed to CPLEX (running on a 1.5GHz PIII Linux machine with no other CPU-consuming tasks), and if the optimum was not found then the current feasible solution and the lower bound were reported as the interval where the optimum lies. Note that the RSNE algorithm always outperforms the shortest path routing, and when relatively large networks are considered the CPLEX integer solver needs much more than 10 minutes in order to find the optimum (we let a 12-node test run for three days without improving the estimate before the branch-and-bound tree caused a memory overflow).

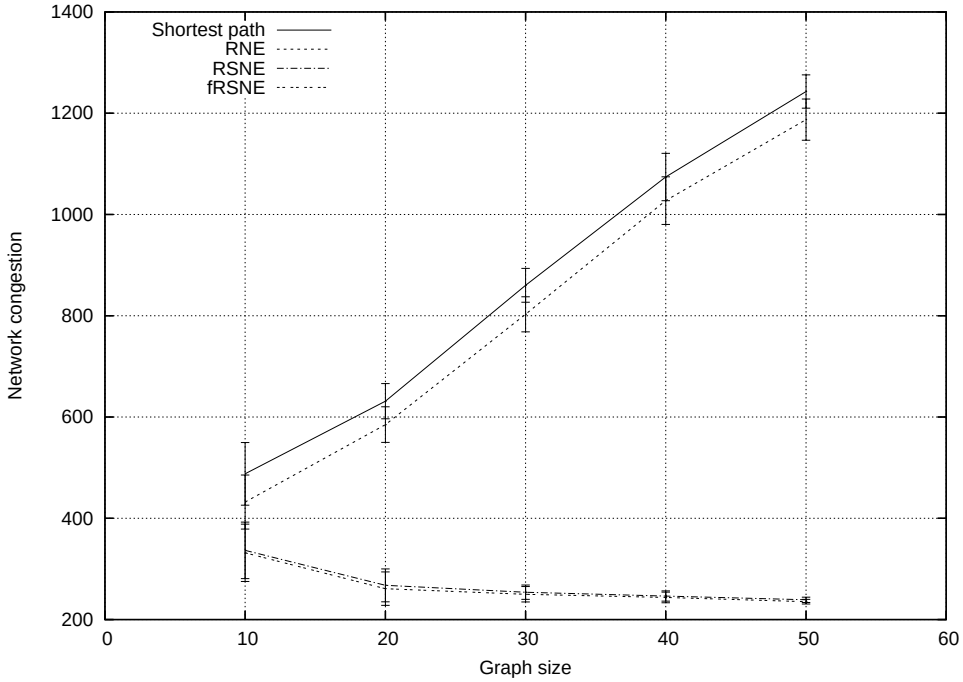


Figure 4.6: Congestion on random networks of different node size, 50% edge density. Random bars represent the 95% confidence interval, the two lower plots, representing `RSNE` and `fRSNE`, are almost equal.

To allow a better comparison among heuristics, a series of experiments were performed on random networks (Figures 4.6 and 4.7) and on Euclidean disk networks (Figure 4.8).

Figures 4.6, 4.7 and 4.8 have been obtained by averaging 50 runs of the algorithms for each plotted bar, representing the 95% confidence interval of the true mean congestion value. In Figure 4.6 node size was varied from 10 to 50 in steps of 10, with a constant 50% edge density, and disconnected networks were discarded. Figure 4.7 compares algorithms on random networks with a constant size of 20 nodes and different densities from 40% to 95%. Figure 4.8 has been obtained from the same experiments on Euler disk networks with radius equal to .3 (remember that points are randomly scattered throughout a unit square). All graphs plot mean congestion values. In all cases, the two lower lines, rep-

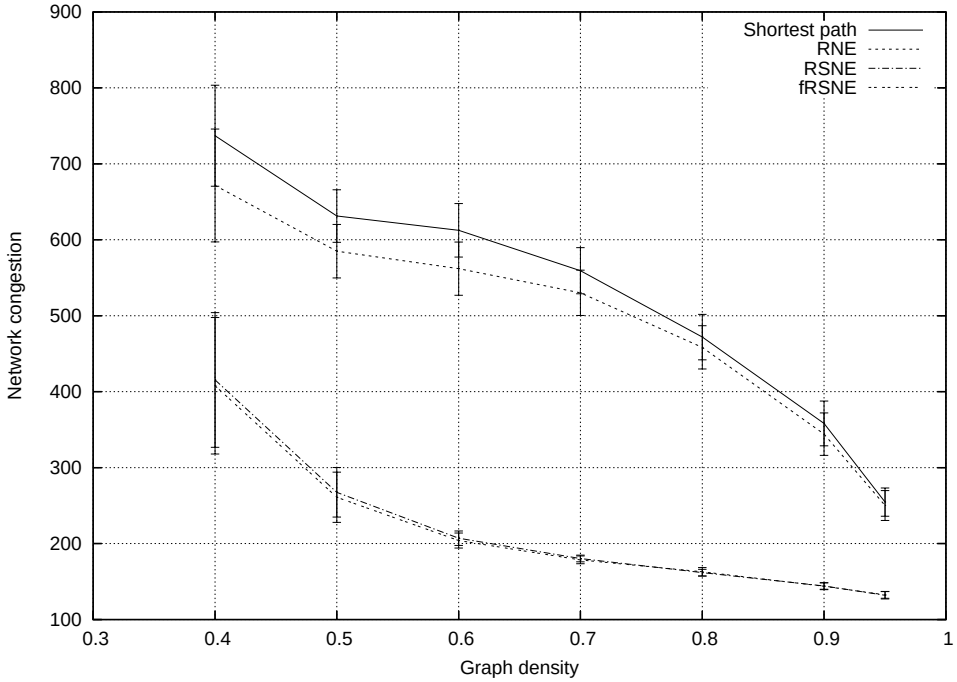


Figure 4.7: Congestion on random networks of different densities, 20 nodes.

resenting RSNE and fRSNE, are almost equal, and this motivates support of the randomized version. On the other hand, the simplified RNE scheme does not show a significant performance improvement over shortest path routing in the random network case. Its use, however, may be motivated in the Euler disk case, where an average 43% congestion reduction is obtained in the 50-nodes case.

Figure 4.6 plots an interesting behavior: while edge congestion obtained by RSNE (or equivalently fRSNE) is almost constant, the shortest path result is linearly increasing. The same can be seen in Table 4.1. This can be explained by considering that both the overall traffic requirement and the number of edges increase as the square of the number of nodes. Techniques aiming at congestion reduction are able to exploit this fact, while shortest path policies, which do not aim at congestion reduction as their primary target, tend to crowd paths along eventual shortcuts. While congestion can be drastically decreased (up to 5.5

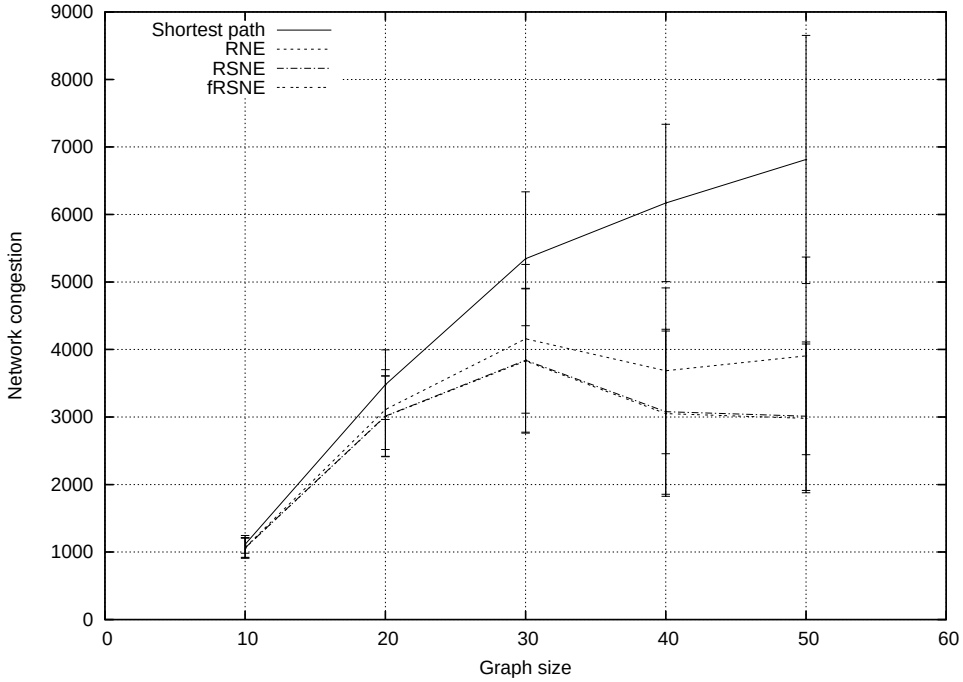


Figure 4.8: Congestion on Euler disk networks of different node sizes, radius .3 on a unit square uniform scattering.

times on 50 nodes), the average hop length increases up to 4% for RSNE and 5.7% for fRSNE. Likewise, increase in average edge load has been detected up to 3.6%.

In Figure 4.7, the congestion reduction operated by RSNE and fRSNE reaches its maximum (67%) at 70% edge density, then it becomes less and less significant, and the different plots tend to the same congestion value as density approaches 100%. This is obvious, as all policies will use one-hop routing on a clique. For the same reason, all average hop lengths tend to 1 as density approaches 100%. The highest increase (3.6% for RSNE, 6% for fRSNE) has been detected for the lowest considered density, 40%.

Figure 4.8 depicts a more realistic situation in which connection depends on distance. In this case confidence intervals are much larger because topology is more variable. Clusters of nodes with few congested inter-cluster links can

appear with significant probability, as well as uniformly distributed networks with balanced loads. Also in this case the highest congestion reduction operated by RSNE and fRSNE (56%) is detected for 50 nodes. The average hop length is increased in the worst case by 12% (from 2.6 to 2.92 for 50 nodes), while the average load is 11% higher.

In all cases considered in this section, a large improvement on congestion reduction has been obtained at the expense of a slight degradation of the other performance indices that had been considered, average hop length and average load.

4.4.4 Tests on dynamic traffic

The 24-nodes network already considered for static traffic has been tested with traffic evolving in time. The traffic evolution pattern is that discussed earlier. The complete time span is of 1000 time steps, with a new independent matrix every 20 steps and linear interpolation on intermediate matrices.

Figure 4.9 reports the results of the simulation for the first 100 steps. As the shortest path outcome is highly variable, we chose to calculate 50 shortest path routings with random tie-breaking per time step. All runs were executed over the same traffic pattern. Error bars represent the span from the lowest to the highest congestion value obtained. The other lines represent one run of RSNE with full restart and 100 iterations at each time step, one run of I-RSNE(1) with just one optimizing iteration per time step and one run of I-fRSNE with one optimizing iteration per time step and all parameters set to 1. As expected from static tests, results achieved with the proposed techniques are always below the shortest path range. The fact that incremental implementations occasionally outperform the static RSNE depends from the low number of iterations per step. An initial transient can be seen for the first 10 steps, while the incremental algorithms descend from their initial shortest path configurations to lower congestions.

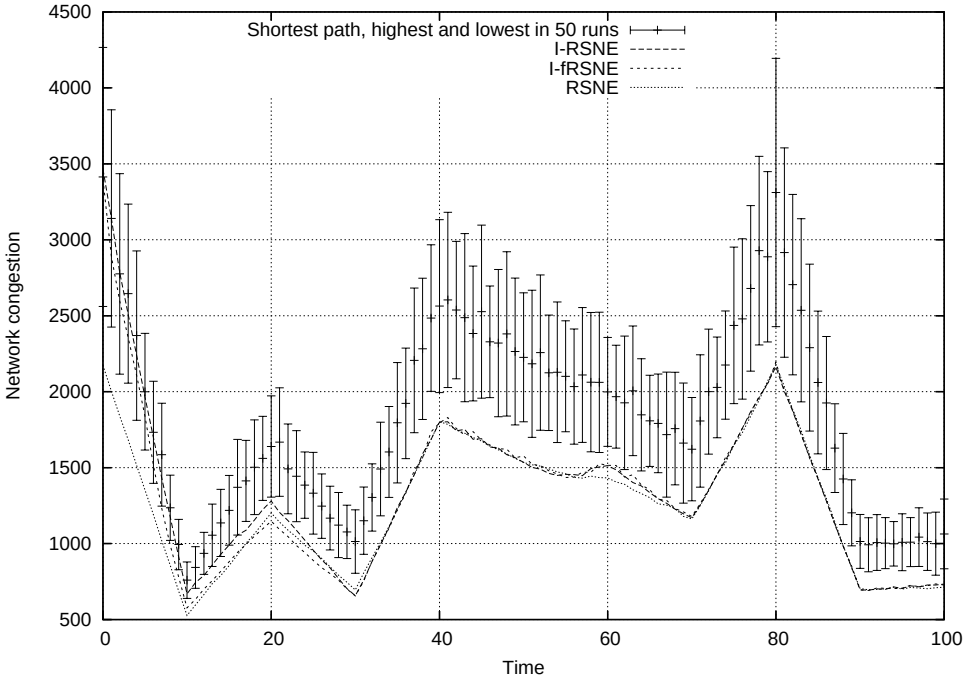


Figure 4.9: Congestion in an online setting.

These results suggest that modifying a single entry in the routing table of a single node at each step is enough to adequately follow the traffic pattern, at least with this traffic model, even with a restricted random exploration of the move space.

While congestion is significantly reduced, a slight increase in the average path length has to be expected. Figure 4.10 represents the behavior of average hop length in the same experiment described above on the complete 1000-step time span. It can be observed that, while all shortest path computations result in the same value (lower horizontal line), and the increase of the static RSNE algorithm remains around the same level (about 5%), the incremental policies tend to suffer from a higher degradation (up to 17.3%) due to progressive abandon of the shortest path configuration. However, this problem can be fixed by triggering a shortest-path restart whenever the average hop length reaches a given threshold, or periodically.

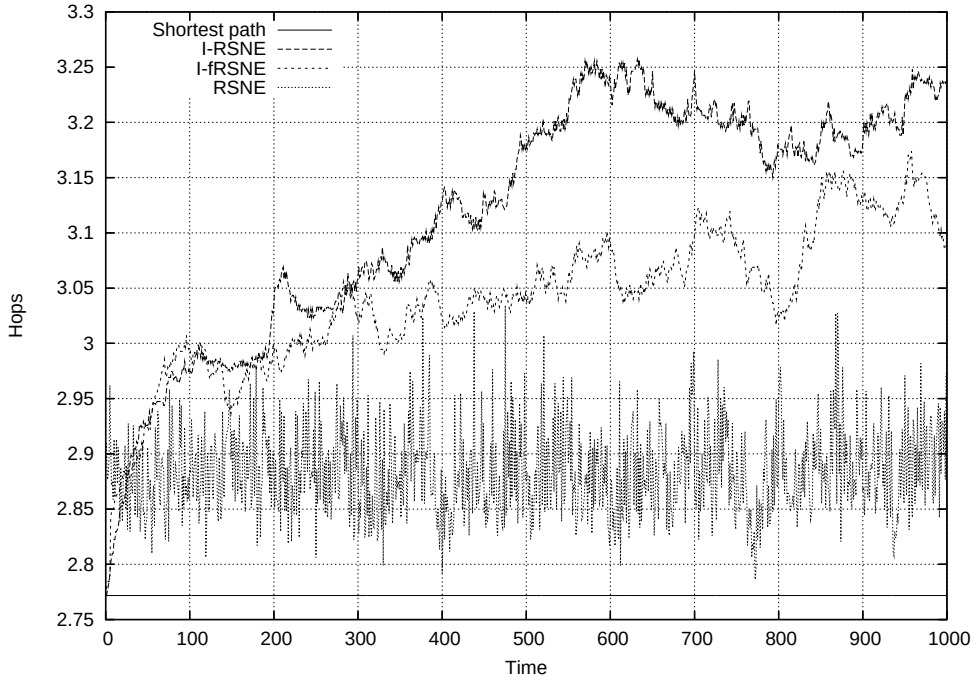


Figure 4.10: Average hop length in a dynamic setting.

4.5 Conclusions

A new Load Balancing algorithms for Optical Networks based on IP-like routing and Local Search has been proposed, where every move modifies a single entry in the routing table of a node. The comparisons between the new scheme and the optimum solutions found via an ILP solver show both calculation time savings and comparable results of network congestion and of average length of the resulting routes. The incremental scheme `I-RSNE` produces congestion values that are almost undistinguishable from those of the `RSNE` algorithm. The randomized scheme called `fRSNE` has a substantially reduced complexity in comparison with `RSNE` and, again, very similar final congestion results.

Chapter 5

Load Balancing Schemes for Congestion Control in MPLS Networks

As described in Sect. 3.2, most of the proposed schemes for congestion control in MPLS networks are preventive, they allocate paths in the network in order to *prevent* congestion.

One of the most cited constrained-based routing (CBR) schemes, called MIRA (Minimum Interference Routing Algorithm) [60], is based on an heuristic dynamic online path selection algorithm. The key idea, but also the intrinsic limitation of the algorithm, is to exploit the a priori knowledge of ingress-egress pairs to avoid routing over links that could "interfere" with potential future paths set-up. These "critical" links are identified by MIRA as links that, if heavily loaded, would make it impossible to satisfy future demands between some ingress-egress pairs. The main weaknesses of this scheme are the computation complexity caused by the maximum flow calculation to identify the "critical" links and the unbalanced network utilization. As Wang et al. demonstrated in [112] with two counterexample topologies, MIRA cannot estimate bottlenecks on links that are "critical" for clusters of nodes. Second, it does not take into account the current traffic load in routing decisions [20]. Let's consider the case where a source-destination pair is connected by two or more routes with the same residual bandwidth. When a new LSP set-up demand arrives, one of these

routes will be chosen to satisfy the request. This implies that after this LSP has been set-up, all the links belonging to the other routes become critical according to the definition given above. This means that all the subsequent requests between the same router pair will be routed over the same route while all the other routes remain free thus causing unbalanced resource utilization.

Furthermore, the preventive behaviour is not sufficient: when LSPs are set up and torn down dynamically, CBR schemes can lead to inefficiently routed paths and to future blocking conditions over specific routes. Therefore, preventive methods are complemented by reactive ones, such as LSP re-routing and LSP bandwidth adaptation.

In this chapter two novel schemes are presented whose main aim is to reduce MPLS network congestion by using load balancing mechanisms based on different Local Search heuristics. The key idea is to efficiently re-route LSPs from the most congested links in the network, in order to balance the overall links load and to allow a better use of the network resources.

5.1 Problem definition and system model

The considered network consists of n routers. In MPLS terminology connections are set-up between an ingress-egress pair of routers. Each connection request arrives at an ingress router (or at a Network Management System in the case of a centralized route computation) which determines the explicit route for the LSP according to the current topology and to the available capacities at the IP layer. It is assumed that every router runs a link state routing protocol with extensions for link residual bandwidth advertisements.

A connection request i is defined by a triplet (i_i, e_i, b_i), where i_i and e_i specify the ingress and egress routers and b_i indicates the amount of bandwidth required. In the rest of the chapter we will consider only the routing of bandwidth guaranteed connections. As in [60], we suppose that Service Level Agreements

(SLA) are converted into bandwidth requirements for the LSP. We assume also an on-line context with connection requests arriving one at a time.

The corresponding LSPs will be routed through the network and, at each instant, one determines the *virtual load* of a link by summing the bandwidth b_i of the connections passing through the link. The difference between the link capacity and the virtual load gives the *residual bandwidth*. The minimum residual bandwidth over all links of a network is called the *available capacity* of the network. This value identifies the *most congested links*.

Given an MPLS network with connection requests arriving dynamically, the objective of our on-line algorithm is to balance the allocation of the already established LSPs in the network to reduce the rejection probability for future traffic demands.

5.2 A Load Balancing Algorithm for Traffic Engineering

We consider algorithms that are based on a sequence of small steps (i.e., on local search from a given configuration) because global changes of the routing scheme can be disruptive to the network. A similar approach has been proposed in papers about Virtual Topology Reconfiguration in optical networks (see Sect. 3.1.3 for an overview). For each tentative move, the most congested links are identified and one of its crossing LSPs is rerouted along an alternate path.

The proposed scheme is similar to the congestion control algorithm RSNE presented in Chapter 4. While in RSNE the search for an alternate route is performed for all the connections crossing the most congested links, the scheme proposed here is based on a faster local search technique: the local search stops as soon as first improving alternate route for one of the LSPs is found, thus dramatically reducing the computational time.

Two different versions are considered, according to the triggering mecha-

nism used to start the load balancing routine:

1. First-Improve Dynamic load balancing, called FID(x). The parameter x indicates the threshold for the link residual bandwidth measured as a fraction of the link capacity, which determines when a link is considered congested. Each new request is routed with WSP (Widest-Shortest Path), a modified Shortest-Path algorithm which runs on a graph where link weights are defined as $w_l{=}1/c_l$, where c_l is the residual available link capacity, ∞ if c_l is zero [53]. After routing, if less than x residual bandwidth is left on some link, the dynamic load balancing algorithm is executed. As soon as the alternate route for one LSP has more residual bandwidth than the original route, the search stops and the LSP is rerouted. If the search cannot find any improving alternate LSPs in the network for all congested links, rerouting is not performed.

2. Lazy First-Improve Dynamic load balancing, called LFID. In this version, the dynamic load balancing is activated when a new LSP request arrives that cannot be satisfied. Now only links having the smallest residual bandwidth are considered congested. The algorithm goes through LSPs crossing them until any improving alternate route computed with WSP is found. If the search is successful, the LSP is rerouted over the path found and another attempt is made to establish the new LSP request. If it fails, the new request is rejected. In [58] the triggering mechanism is similar but the authors used ILP for reallocating the LSPs.

Figure 5.1 shows the pseudo-code of the load balancing algorithm. Let us first define the notation. The most congested links are collected in the $congestedLinkSet$ which is computed through the function $calculateNetwork\text{-}Load$. This function is different for the two implementations of the algorithm. In FID(x), the most congested links are all the links whose residual bandwidth is lower than the fraction x of their capacity. In LFID, the most congested links

```
1. <congestedLinkSet> ← calculateNetworkLoad
2. betterLSPFound ← false
3. while (congestedLinkSet ≠ ∅) and (not betterLSPFound)
4.     (cFrom, cTo) ← pickElement (congestedLinkSet)
5.     LSPSet ← all the LSPs crossing link (cFrom, cTo)
6.     while (LSPSet ≠ ∅) and (not betterLSPFound)
7.         LSP_i ← pickElement (LSPSet)
8.         currResBdw ← residual bandwidth on the LSP_i's route
9.         removePartialLoad (LSP_i)
10.        find an alternate path A_LSP for LSP_i
11.        if (A_LSP is found)
12.            altResBdw ← residual bandwidth on the A_LSP's route
13.            if altResBdw > currResBdw
14.                betterLSPFound ← true
15.                oldLSP ← LSP_i
16.        restorePartialLoad (LSP_i)
17. if (betterLSPFound)
18.    reroute traffic from oldLSP to A_LSP
```

Figure 5.1: The First-Improve Dynamic load balancing

are all the links with the minimum (relative) residual bandwidth at the moment. The advantage in this case is that we do not need to set a threshold, which is a critical parameter and depends on the traffic load in the network.

For each iteration cycle, we consider each congested link in $congestedLinkSet$, identified by its endpoints ($cFrom$, cTo). For each LSP_i crossing the link ($cFrom$, cTo), taken in decreasing bandwidth request order, one tries to reroute it on an alternate route. This move is accepted only if the new path increases the available capacity of the network, calculated as the minimum (absolute) residual bandwidth available on the route of LSP_i, which is $currResBdw$. To perform this operation, one temporarily removes the load of LSP_i from the current link, and finds a new path (A_LSP) using WSP starting from the ingress LER (I-LER) which originated the LSP itself, provided that all the links considered congested ($congestedLinkSet$) are avoided. If an alternate path for LSP_i is found, the minimum (absolute) residual bandwidth available on it is calculated as $altResBdw$ and compared to $currResBdw$. If the available capacity of the

network is improved, the move is accepted and in the last part of the algorithm the rerouting is executed (lines 17–18).

Let us consider the worst-case computational complexity. The proposed algorithm is composed of nested cycles. Let n be the number of nodes in the network and m its number of links. The number of iterations for the loop at line 3 is only bounded by the number of links in the network, while for the loop at line 6 is bounded by the number of LSPs that cross the congested link, called k. The computation of the alternate path for the selected LSP_i using Dijkstra' shortest-path algorithm requires $O(nm)$. This can be improved to $O(nlogn + m)$ by using a priority queue with Fibonacci heap in the implementation. Functions $removePartialLoad$, $restorePartialLoad$ and calculations of residual bandwidth on LSP route have complexity at most $O(n)$, since an LSP cannot contain more than n hops. All the other functions require a constant computational time. The value of k depends on the average bandwidth of the LSPs in the network and the link capacities. Therefore the complexity of our algorithm is not more than $O(knm^2)$. As demonstrated in the experimental section, the actual empirical complexity is much lower than this worst-case bound.

5.3 Simulation results

The simulations are carried out by using the network topology of [60], see Figure 5.2. The links are all bidirectional with a capacity of 120 units (thin lines) and 480 units (thick lines). These values are taken to model the capacity ratio of OC-12 and OC-48 links. In order to compare our schemes with MIRA, traffic requests are limited only to the ingress and egress router pairs (S_1,D_1), (S_2,D_2), (S_3,D_3) and (S_4,D_4). However, it is important to highlight that our algorithms allow to relax this strong constraint.

The experiments compare our algorithms with Minimum-Hop Algorithm

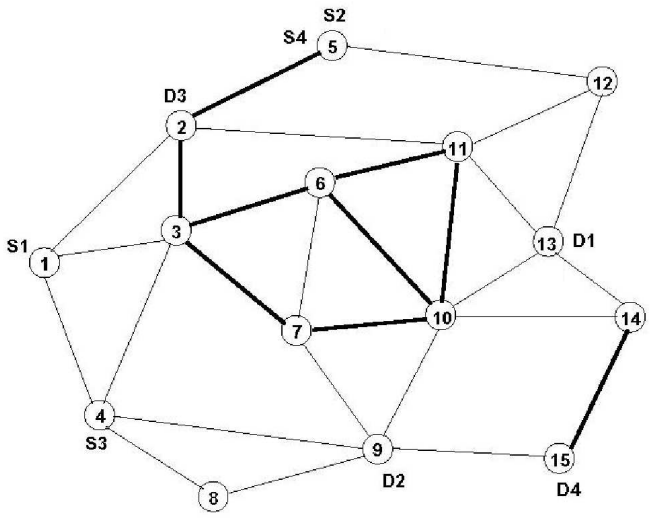


Figure 5.2: The network topology used in the simulations.

Table 5.1: Blocking probability (and improvements w.r.t. MHA)

Algorithm	Bw $\in \mathcal{U}[1,3]$, $\lambda/\mu = 150$		Bw $\in \mathcal{U}[1,6]$, $\lambda/\mu = 75$	
MHA	$12.5 \cdot 10^{-2}$	-	$8.0 \cdot 10^{-2}$	-
MIRA	$9.4 \cdot 10^{-2}$	(–25.1%)	$3.9 \cdot 10^{-2}$	(–50.7%)
FID(0.01)	$8.2 \cdot 10^{-2}$	(–34.6%)	$3.3 \cdot 10^{-2}$	(–58.1%)
LFID	$8.6 \cdot 10^{-2}$	(–31.4%)	$3.3 \cdot 10^{-2}$	(–58.1%)

(MHA) and Minimum Interference Routing Algorithm (MIRA). Connection requests arrive between each ingress-egress pair according to a Poisson process with an average rate λ, and their holding times are exponentially distributed with mean $1/\mu$. Ingress and egress router pairs for each LSP set-up request are chosen randomly. The network is loaded with 20000 requests during one trial.

The first results (Table 5.1) show the blocking probability of MIRA and both implementations of our algorithms FID(x) (with $x = 0.01$) and LFID, comparing them with MHA. The rejection ratios shown are average values calculated over 10 runs. The first experiment considers the same traffic distribution used by Kodialam et al. [60], i.e. bandwidth demands for LSPs uniformly distributed

between 1 and 3 units and $\lambda/\mu = 150$ for each ingress-egress router pair. MIRA and the proposed algorithms perform all better than MHA, but while LFID perform slightly better than MIRA (with a 6% decrease in blocking probability), FID(0.01) performs the best (10% ca. over MIRA). The second experiment considers LSPs with higher capacity on average, i.e. the bandwidth demands are uniformly distributed between 1 and 6 units, but considering half the λ/μ ratio compared to the previous case. MIRA and the proposed algorithms perform all better than MHA, while both FID(0.01) and LFID perform similarly, showing a 7% decrease in blocking probability with respect to MIRA.

These first results show that our algorithms perform slightly better than MIRA, especially if the traffic considered is characterized by bandwidth-consuming LSPs. This can be explained by the implicit mechanism of First-Improve Dynamic load balancing algorithm, which reroutes an LSP away from the most congested link and considers first the LSPs with higher bandwidth request, thus guaranteeing a faster network congestion reduction.

In order to evaluate in more detail the proposed algorithms, different set of experiments have been performed. The first set considers a uniform distribution of traffic among all the ingress-egress pairs (symmetric traffic). The second set of experiments considers a non-uniform distribution (asymmetric traffic). In particular, it is assumed that ingress-egress pair (S_1,D_1) generates a traffic rate which is four times higher than the other pairs, on average. This set allows us to highlight MIRA limitations regarding the traffic load condition over the MPLS network. Two experiment subsets are performed for two different bandwidth distribution per LSP (maximum bandwidth 3 and 6).

Figure 5.3 presents the results for symmetric traffic. Two different bandwidth distributions per LSP are considered: the first has a maximum bandwidth of 3 units, while the second 6 units, thus simulating the case of bigger connections on average. Rejection values are calculated over 10 runs: the error-bars are not shown in the plots because they are hardly visible, in the order of 4%. Our algo-

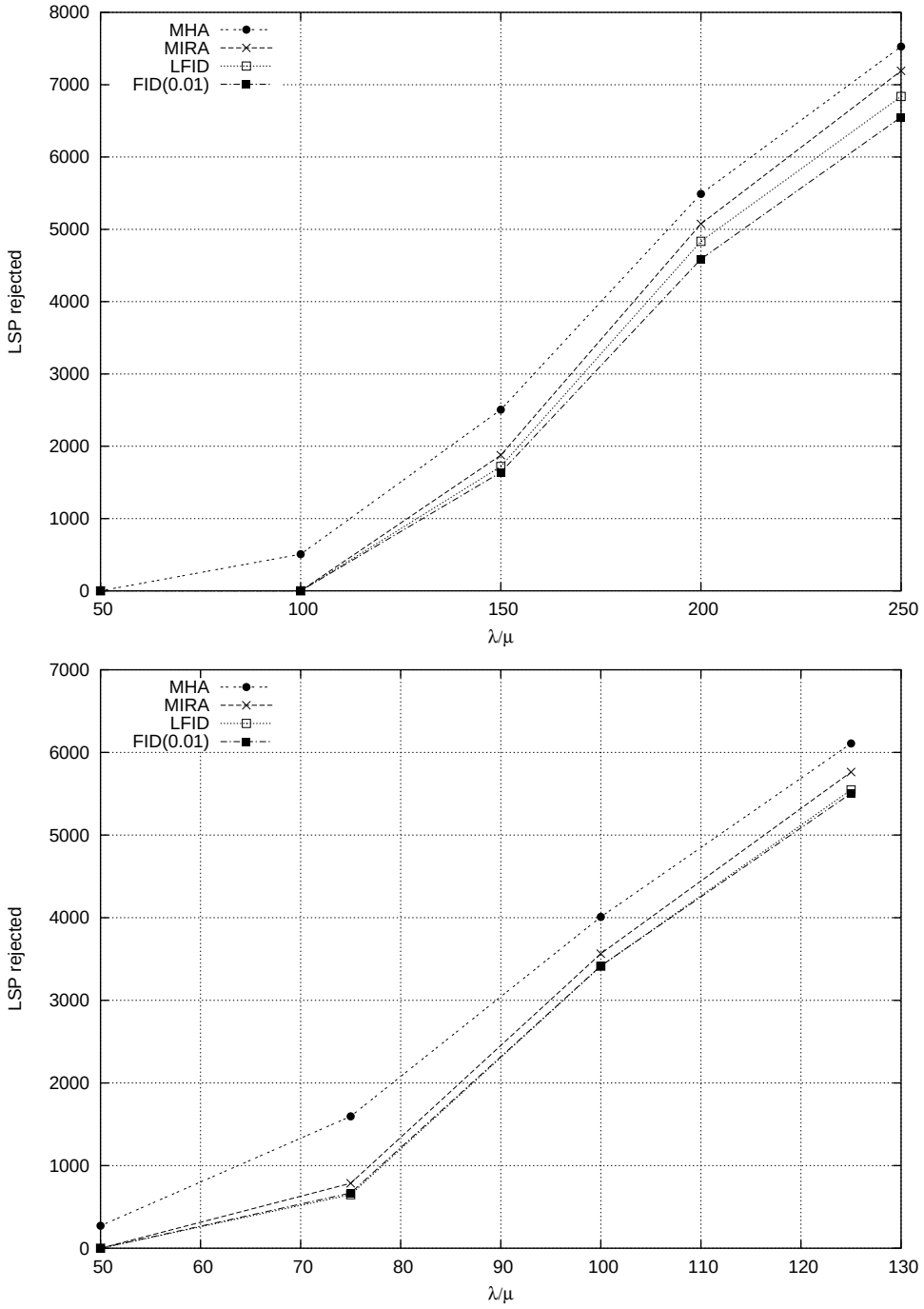


Figure 5.3: Symmetric traffic: number of rejected LSPs vs. λ/μ for maximum bandwidth equal to 3 (upper plot) or equal to 6 (lower plot)

Table 5.2: Performance of the proposed reactive schemes

Algorithm	LSPs rerouted	Routes computed	Num. of congLink	Num. of LSPs per congLink
FID(0.01)	8152.7	1107K	3.6 ± 2.7	61.8 ± 6.4
LFID	1368.5	260K	3.3 ± 2.2	62.1 ± 3.8

rithms perform slighlty better than MIRA in these traffic conditions. The upper plot shows that FID(0.01) performs slightly better than LFID for high values of λ/μ, due to the implicit reaction mechanism: the load balancing algorithm is triggered whenever some link overcomes the congestion threshold, thus keeping a better distribution of the load over the network.

Figure 5.4 presents the results for asymmetric traffic. Two different bandwidth distributions per LSP are considered as above. The plots show that these traffic conditions lead to a higher blocking probability on average compared to the previous set of experiments. Capone et al. [23] proved that MIRA does not perform well when asymmetric traffic is applied to the ingress-egress pairs. The proposed schemes perform much better than MIRA mainly thanks to the independence of our algorithms from the traffic conditions. As in the first set of experiments, FID(0.01) performs slightly better than LFID.

Table 5.2 shows a comparison between the two proposed schemes FID(0.01) and LFID over 20000 LSP requests, by using symmetric traffic with bandwidth demands for LSPs uniformly distributed between 1 and 3 units and λ/μ = 150. The first column shows the number of LSPs rerouted during the algorithm run, while the second column shows the number of alternate routes considered during the simulation before finding the best path to reroute, a measure of the computational time spent by the algorithm. From these values it can be noticed that, even if it is the best performing scheme in term of rejected requests, the FID(0.01) scheme leads to the highest number of re-routings in the network

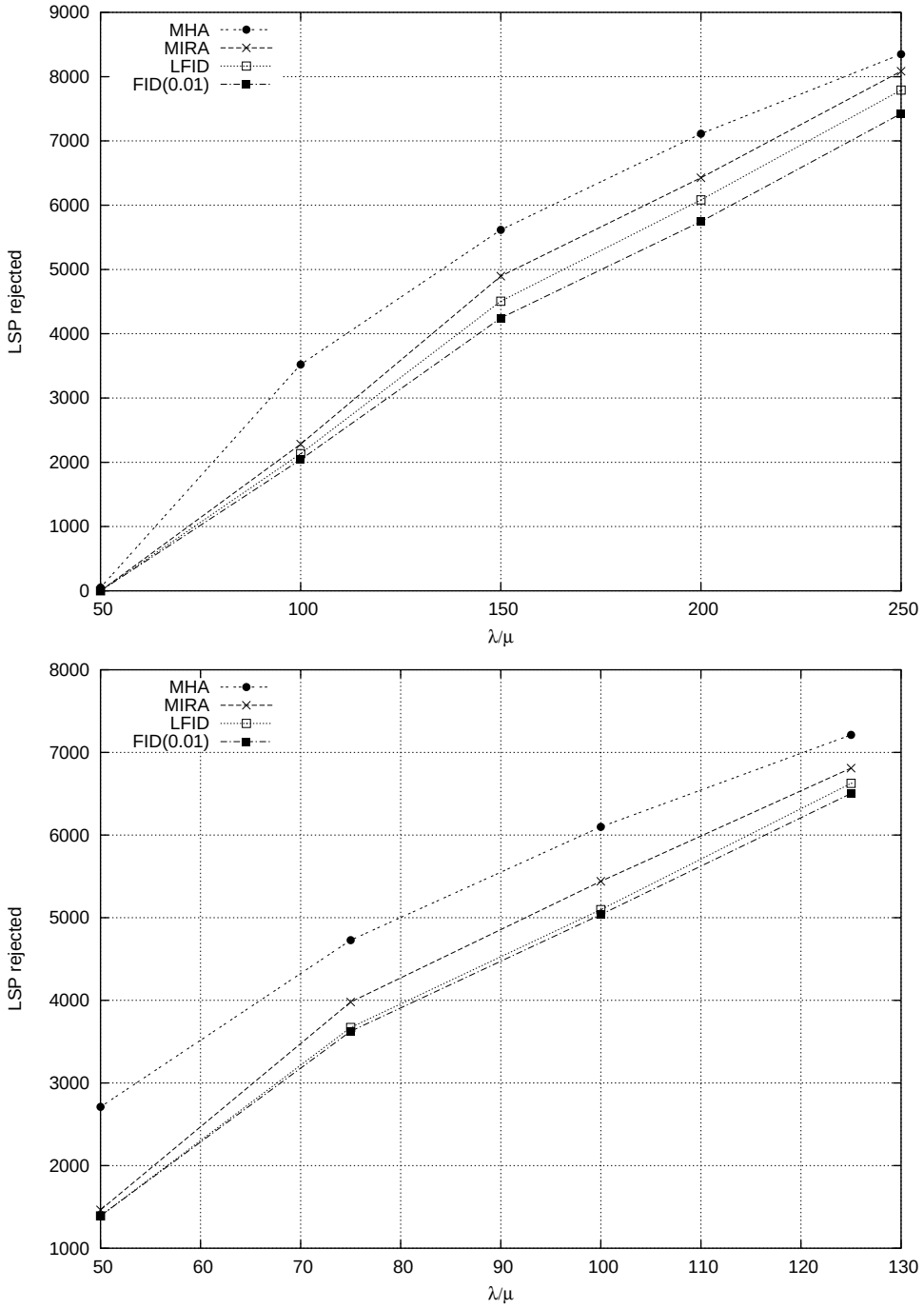


Figure 5.4: Asymmetric traffic: number of rejected LSPs vs. λ/μ for maximum bandwidth equal to 3 (upper plot) or equal to 6 (lower plot)

and requires more computation. In addition to requiring less computation and reroutings, LFID has the additional advantage of being independent from the threshold value. The third column shows the average number of links considered congested by our algorithms during their run, while the fourth column shows the number of LSPs crossing each congested link on average. These values prove that the number of links considered congested is always a small percentage of the total number of links (for the specific network we are considering in our experiments, it is between 2% and 12%). This means that the empirical computational complexity is much lower than the worst-case bound calculated in Sect. 5.2.

The results demonstrate that it is possible to obtain better rejection ratios with the proposed reactive schemes than with MIRA and at a reduced computational cost. In fact, when a MIRA CBR scheme is employed, each time a new LSP needs to be routed over the network, the maximum flow between each ingress-egress pair nodes in the network has to be computed in $O(n^2\sqrt{m})$ before applying the traditional $O(nm)$ shortest-path algorithm [60]. In our reactive scheme instead, each LSP is installed directly by using Dijkstra algorithm. The load balancing algorithm is triggered only when network congestion is detected, and this happens for a small percentage of LSPs which need to be rerouted. In particular for LFID, the proposed algorithm is computed only for 7.5% of the total number of LSPs, see Table 5.2.

5.4 Conclusions

Two new online reactive schemes to perform Traffic Engineering in MPLS called FID(x) and LFID have been presented. Simulation results show that our algorithms perform better than Minimum Interference Routing Algorithm (MIRA), a well-known scheme in literature to prevent congestion in MPLS networks, by reducing both the LSPs rejection probability and the computa-

tional complexity. Both of them show their best results when the traffic inserted into the network is characterized by high-capacity and long-lived connection requests. In fact, in the case of a bandwidth-hungry LSP that lasts for a long time, there is a higher risk of blocking future connection requests due to a sub-optimal allocation. A reactive congestion control scheme behaves better than a preventive scheme, because it can dynamically adjust the inefficiently routed paths. Among the two proposed schemes, LFID has the advantage to minimize the number of LSPs to be rerouted in the network to balance the traffic and therefore has the lowest computational time and the lowest disruption of traffic.

Chapter 6

A Traffic Engineering scheme for QoS routing in IPO Networks

While a significant amount of research has been done on guaranteeing Quality of Service in pure IP-based networks, the problem of providing QoS guarantees to different services carried over high-capacity optical channels remains largely unsolved for wavelength-routed IPO networks [59]. In the literature, the concept of "Quality of Service" in such networks assumes two main meanings: *service differentiation* or *transmission quality*, seldom jointly considered. In the first case most papers propose Routing and Wavelength Assignment (RWA) algorithms which allow to dynamically assign a set of wavelengths to higher-priority traffic in order to maximize the revenue for the service provider [57, 27, 89]. In the second case the transmission impairments introduced by the physical layer are considered in the RWA algorithms, which assign optical paths to incoming requests only if the resulting lightpath is feasible (i.e. the output SNR is good enough to guarantee the required transmission quality to the bitstream). In this case no service differentiation is considered because the same QoS is guaranteed for all the carried traffic [119, 91, 11].

All these proposals consider the routing of an entire lightpath (RWA problem). When an optical network with Traffic Grooming capabilities is considered, most of the previous works are focused on optimizing the physical re-

sources usage by considering specific constraints on the optical node architecture, as described in Sect. 3.3. Very little attention has been paid so far on the effects of both the physical constraints characterizing the optical layer and the delay restrictions of a multi-hop path over the traffic carried on the wavelengths. If for example some packet-loss sensitive traffic (e.g. real-time video) is carried over a lightpath experiencing strong transmission impairments, the resulting SNR (Signal-to-Noise Ratio) degradation could be so high to compromise the signal quality requirements. If a connection carrying delay-sensitive traffic (e.g. Voice-over-IP) is routed over a multi-hop path, the output signal could suffer very high end-to-end delay due to the optical-to-electronic (o-e-o) conversions and queueing delays experienced along the path.

To guarantee the specific requirements of different applications, a "QoS-aware" dynamic grooming scheme should then consider these constraints when a new connection must be routed. Unfortunately this mechanism could not be sufficient, because requests with tighter requirements (usually the more profitable from the Service Provider viewpoint, thus with High Priority - HP) are likely to be blocked by the requests with less or no QoS requirements (Low Priority - LP). In order to guarantee lower blocking probability to higher priority classes, different mechanisms can be applied. In [111] a dynamic grooming algorithm which admits high-priority (HP) requests in preference over the low-priority ones is proposed, by implementing an admission control mechanism which guarantees a limited portion of resources (in term of ports number) to low priority traffic. The main disadvantage is how to decide the right threshold for the resources dedicated to HP traffic. An MPLS-based preemption scheme to deal with different traffic classes is used in [19], but the main drawback is that class priorities are not mapped onto the optical layer constraints, thus HP traffic can be routed over bad paths in term of delay and packet-loss.

In this chapter a set of minimal QoS requirements for a high-priority class of service is defined first at the IP level, and then the corresponding QoS con-

straints are defined at the optical level. Then a novel Traffic Engineering (TE) scheme is presented, based on two concurrent heuristics: a dynamic grooming algorithm, which routes incoming connection requests to guarantee their QoS constraints in term of transmission quality, and a preemption mechanism, which provide service differentiation by minimizing the blocking probability for HP requests.

6.1 QoS requirements on groomed traffic

The network architecture considered is IP over Optical based on a *peer* interconnection model. Two node architectures are considered here (see Sect. 3.3): a node can be a Non-Grooming OXC, which allows to switch entire lightpaths from an ingress port to an egress port, or it can be a Multi-hop full Grooming OXC, where the electronic fabric is based on a IP/MPLS router. These nodes can both terminate transit traffic or they can groom it into some optical pipe with incoming IP traffic. In the rest of this chapter the terms (IP) router, LSR (Label Switching Router), and G-OXC are used interchangeably. If cross-connects have wavelength conversion capability, we assume they use electronic wavelength converters only. Let R be the set of G-OXCs, T the set of (Non-Grooming) OXCs without wavelength converters and S the set with wavelength converters.

In this architecture, a path connecting two routers in the IP layer is called a *virtual* or logical path, because it is created over some established lightpath in the optical layer. IP traffic dynamically follows the virtual topology built by the optical level underneath. A G-MPLS like control protocol is assumed (see Sect. 2.3.3), so that each node is always informed of the network status in term of wavelength usage and lightpath occupation.

A connection request is defined by a quadruplet $D(s, d, b, q)$, where s and d specify the ingress and egress nodes, b indicates the amount of bandwidth

required and q specifies the Class Type (CT), by using the MPLS terminology to aggregate classes of traffic with similar requirements [45]. In the rest of the chapter we consider only the routing of bandwidth-guaranteed IP flows requests, which are carried over Label Switched Paths (LSP) set up through an MPLS-based signalling plane such as RSVP-TE.

The decision to route incoming requests over the existing virtual topology or to establish new lightpaths to create more room for them can lead to different network performance. As a result, a request can be routed over a direct lightpath (a *single-hop* path at the IP level), if it crosses only pure OXCs between an ingress and an egress router, or over a sequence of lightpaths (a *multi-hop* path at the IP level), if it crosses intermediate LSR nodes. Furthermore, a network operator should take into account the specific QoS requirements of the incoming request when deciding its route along the network. In fact, a connection request routed over a single or multi-hop path in the virtual topology would experience different *delays* or *packet losses* according to the physical characteristics of the optical pipe carrying the request.

In the following, the impact of these two parameters is considered in more detail, and some specific constraints are highlighted when specific kind of traffic needs to be routed in the network:

Delay. Most of delay suffered by a traffic flow derives from the queueing delays in IP routers[1] and from optical-to-electronic-to-optical (*o-e-o*) conversion delays in regenerators and electronic wavelength converters [111]. Then we assume that, when some *delay sensitive* application needs to be routed in the optical network, the following constraint must be applied: the corresponding connection request cannot be carried over optical pipes consisting of more than $C_{max} + 1$ lightpaths, i.e. it can't experience more than C_{max} o-e-o conversions.

Packet losses. The transmission impairments that digital transmission experi-

[1] In [84] it has been proved in fact that in a network implementing WFQ (weighted fair queueing) scheduling with leaky bucket traffic shaping at ingress nodes, the maximum packet delay a request can tolerate is proportional to the number of crossed routers.

ences along a lightpath without intermediate electronic regeneration can impact the packet loss ratio of the connection carried over it. In fact, ASE (amplified spontaneous emission) noise in optical amplifiers, insertion loss and crosstalk introduced by OXCs and attenuation and PMD (Polarization Mode Dispersion) effects introduced by the fibers can degrade the optical signal resulting in a very high BER (bit-error rate) [108]. In the rest of the chapter we assume the simplified hypothesis that all the fiber links introduce the same level of transmission impairments[2], thus reducing the problem of selecting a good lightpath for packet-loss sensitive traffic to the problem of limiting the maximum number of hops (up to H_{max} fiber links) for the lightpath which carries such traffic.

In the rest of the chapter we consider two Class Types only: a high-priority (HP) class, characterized by minimum end-to-end delay and low packet loss probability (e.g. high-quality real-time services), and a low-priority (LP) class with no QoS requirements, which can experience both high end-to-end delay and frequent retransmission when routed over lightpaths with higher BER or if disrupted due to rerouting (e.g. classical best-effort traffic).

Given an IPO network with connection requests belonging to different Class Types arriving dynamically, the objective of our on-line TE scheme is twofold: first, it must route the request according to the specific QoS requirements and second, it must balance the allocation of the already established LP connections in the network to maximize the success probability for HP traffic demands.

6.1.1 A layered graph representation

The optical network is modelled as a layered graph as in [63], where wavelengths on a link are separated into different graph edges. The routing algorithm used to dynamically route the connection request works on a graph which

[2]This assumption can be relaxed by considering a more realistic network such as in [91], but at this stage we believe it is reasonable enough to study the specific problem of guaranteeing different QoS requirements to sub-wavelength connection requests.

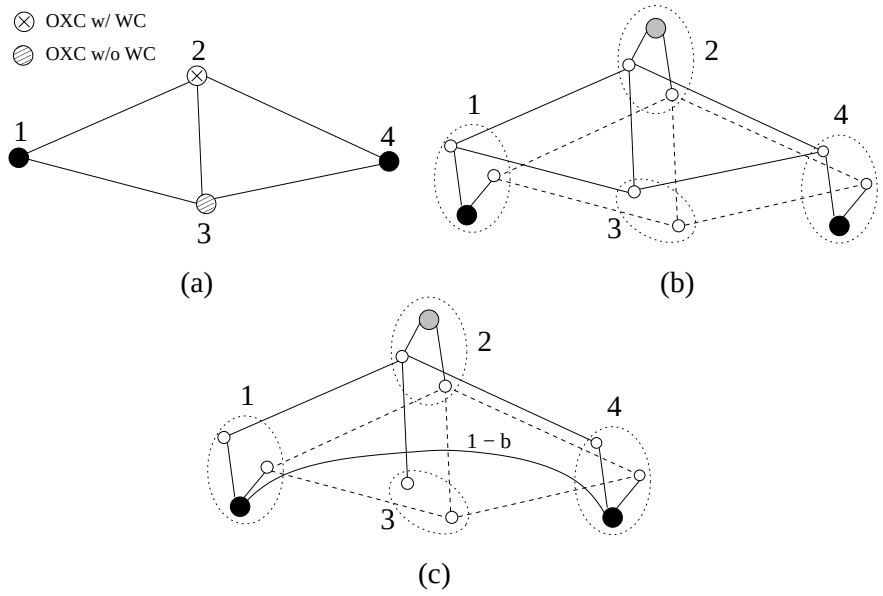


Figure 6.1: The layered graph representation of an optical network

is modified after every successful connection.

In Figure 6.1 a network with two wavelengths per fiber is considered: the *extended graph* $\mathcal{G}$ in (b) is obtained by expanding each node in the network (a) into a number of sub-nodes, one per wavelength, and then by connecting each sub-node to a wavelength on each incoming and outcoming link. Let define as $\mathcal{E}$ the set of edges of this graph. G-OXCs are represented adding some *supernodes*, that topologically connect all lambda layers through "fictitious" links with infinite capacity.

The extended graph allows to model the wavelength availability per link and the residual bandwidth per logical link at the IP layer. In the considered example, the initial (full) capacity of each edge is normalized to 1. As soon as the routing algorithm finds a path between an ingress-egress pair in $\mathcal{G}$ (e.g. from node 1 to 4 through 3 in Figure 6.1 (c)) we modify $\mathcal{G}$ by removing the edges traversed by the lightpath and by adding a direct edge, called *cut-through* with capacity equals to $1 - b$. When an established lightpath is torn down because the last connection occupying it is ended, the cut-through arc is removed and

the edges in the extended graph corresponding to the underlying physical links are set back with full capacity[3].

Different grooming policies can be realized by modifying the weights of the edges in $\mathcal{E}$ and then by running a shortest-path algorithm on the extended graph. These weights reflect the cost of network elements such as o-e-o converters (routers) or free wavelength on some link.

Three possible kind of edges can be identified in $\mathcal{E}$, each having a property tuple $P(c, w, h)$, where c is the edge capacity (if g is the wavelength capacity in bandwidth units, $c = g$ means full capacity), w the associated weight and h the cost metric which models the signal degradation introduced by the transmission link:

- *Wavelength Edges (WE).* An edge $e \in \mathcal{E}$ from node i to j on wavelength layer k is a WE if there is a physical link from i to j and wavelength λ_k is free on this link. For such an edge: $c = g$ and $h = 1$.

- *Lightpath Edges (LE).* An edge $l \in \mathcal{E}$ from node i to j on wavelength layer k is a LE if there is a direct lightpath (cut-through) from i to j on wavelength λ_k. For such an edge: $c = g - \sum_{i=1}^{M} b_i$, if M connections (LSPs) with bandwidth b_i are running over it ($b_i < g$), and $h = H_l$, if the lightpath crosses H_l fiber links.

- *Converter Edges (CE).* These are all the so-called fictitious edges $f \in \mathcal{E}$ between the super-nodes in $\mathcal{G}$ and the nodes belonging to the wavelength layers. For such an edge: $c = \infty$ and $h = 0$.

6.2 A Traffic Engineering scheme for QoS routing

The proposed Traffic Engineering scheme for QoS routing in IP over Optical networks consists of two main components: a dynamic grooming algorithm

[3]The reader is referred to [63] for further details on these operations.

and a preemption mechanism.

6.2.1 QoS-aware Dynamic Grooming

Many dynamic grooming algorithms for *peer* IPO networks have been proposed recently, see Sect. 3.3.2 for an overview. For all schemes the basic idea is to perform a constraint-based routing algorithm in order to maximize the utilization of the network resources thus minimizing the blocking probability. The QoS requirements are not considered in all the proposed algorithms, thus no attention is paid on the impact of the route selection both on delay and signal degradation, which instead are fundamental when an operator wants to guarantee a certain quality to specific Class Types.

When a new request $D(s, d, b, q)$ arrives in some ingress router, there are four possible operations that can be applied [126]:

- *RouteDirect*: route the traffic onto an existing lightpath connecting directly s to d.
- *RouteVT*: route the traffic over the existing virtual topology.
- *RouteNew*: set up a new lightpath l connecting s to d.
- *RouteMixed*: set up a certain set of new lightpaths, which do not connect s and d directly, and route the traffic through them and some existing lightpaths in order to maximize the usage of the virtual topology.

Each operation can be applied only if some prerequisites are satisfied: for example when no direct lightpaths connect node s to d, *RouteDirect* cannot be applied. When a new request needs to be routed in the network, these operations are applied sequentially in different priority order. Then different grooming objectives can be achieved by modifying the sequence of operations. We are not interested here in studying the impact of different grooming policies on the

overall blocking probability *per se* (as in [126]) but instead in studying their impact over the entire TE scheme to guarantee specific QoS requirements for the incoming requests.

Furthermore, when compared to the operations proposed in [126], here the constraints which characterize high-priority requests lead to some specific differences. For example, when a HP request must be routed, all the LEs l with $h_l > H_{max}$ must be skimmed from $\mathcal{E}$ in both *RouteDirect* and *RouteVT*, while in *RouteNew* the resulting path is accepted only if $h_l \leq H_{max}$. In particular, *RouteMixed* has different implementations according to the Class Type of the request, by reusing the concept of *dominant edges* [126]. For a low-priority request, where the objective is to maximize the existing lightpaths usage, we want to minimize the number of Wavelength Edges (WE) in the path found in $\mathcal{G}$. Hence we just need to assign a large weight to all WEs (make them "dominant") such that the weight of a path containing m WEs (with $m \geq 1$) is always greater than that of a path containing $m-1$ WEs. For a high-priority request instead, the objective is to assign a path which minimizes the number of conversions in the network. Then we should make dominant all the edges of such a path which imply an o-e-o conversion: these are all LEs respecting $h_l \leq H_{max}$ (otherwise they must be skimmed from $\mathcal{E}$) and all WEs which depart from a node belonging to set T to a node belonging to set R or S. Moreover for an HP request, the resulting path is accepted if $h_l \leq H_{max}$ and the number of o-e-o conversions is no more than C_{max}.

Specific QoS requirements are guaranteed to different connection request by considering distinct grooming policies to each Class Type, as shown in Figure 6.2. In particular, every time a new HP request arrives at some ingress router s, we consider two "extreme" grooming policies: `VT-first` and `PT-first`. The arrows in the Figure indicate the sequence of the operations used for each policy. The first always tries to route the request over the existing lightpaths before modifying the virtual topology; while the utilization of the optical re-

For a high-priority request:

- `VT-first`: Maximize the virtual topology usage
  RouteDirect $\Rightarrow$ *RouteVT* $\Rightarrow$ *RouteNew* $\Rightarrow$ *RouteMixed*
- `PT-first`: Maximize the optical resources usage
  RouteDirect $\Rightarrow$ *RouteNew* $\Rightarrow$ *RouteVT* $\Rightarrow$ *RouteMixed*

For a low-priority request:

- *RouteDirect* $\Rightarrow$ *RouteVT* $\Rightarrow$ *RouteMixed*

Figure 6.2: The grooming policies adopted for HP and LP requests

sources is definitely improved, the required electronic processing at intermediate hops is also increased. The second instead increases the logical connectivity at IP level (the resulting virtual topology will be more connected), but leads to a heavy usage of the available optical resources.

When instead an LP request must be routed, we decide to minimize its impact over the network by applying only one grooming policy (see Figure 6.2). In this case a new direct lightpath should be installed only when no other possibilities are available. Note that the last operation includes the possibility to set-up a new direct lightpath as a last option, only when the related path costs less than setting up a mixed path with existing and new lightpaths.

If the incoming request cannot find a path, the traffic must be blocked if it is low-priority, while a preemption algorithm is applied if it is high-priority.

6.2.2 Local Preemption Algorithm

In MPLS the preemption is implemented in a distributed way by using the RSVP signaling protocol. When an LSP having priority p (where $0 \leq p < 7$ and 0 is the highest priority) needs to be set-up, its priority is sent in the PATH message along the route selected in the ingress router. When this message reaches an intermediate core LSR, if the available capacity on the outgoing link is not sufficient to carry the request, a local selection of the lower-priority LSPs to

be preempted is performed by the core router, and then the proper notification messages is sent upstream to the edge routers which should try to reroute (or block) these LSPs. This mechanism can potentially involve all the LSRs in a network to finalize the set-up of one high-priority LSP.

Many distributed algorithms for routers implementing a preemption mechanism have been proposed in literature to select the best connection to reroute, according to different objectives (typically: minimizing the number of reroutings in the network). In [87, 37] the proposed algorithms are optimal with respect to their objective functions: in the rest of the chapter, we indicate them as *global* preemption algorithms (GPA), to highlight the (potential) involvement of all LSRs in the network. Thus from the point of view of the signaling, they could result in a large amount of RSVP messages around the network which increases the amount of overhead. When considering a network based on Generalized MPLS, the complexity of this mechanism grows, because the signaling must take into account much more information regarding the optical layer.

The proposed TE scheme considers a sub-optimal preemption mechanism which is based on a simpler implementation both from the algorithmic and the signaling point of view. In the following, the proposed algorithm is called *local* preemption algorithm (LPA) to distinguish it from the optimal one (GPA). The motivation originates from the property of LSR routers in MPLS to have complete information about the crossing LSPs (each node mantains information about all the LSPs that originate, terminate or cross the node itself [37, 19]). Then, by focusing only on the ingress routers of the network, the simplest sub-optimal preemption mechanism to apply is to perform local selection of one or few low-priority LSPs to preempt when a high-priority LSP is blocked.

Figure 6.3 shows the pseudo-code of the *local* preemption algorithm in a G-MPLS based network, which is triggered in the ingress router s when a high-priority request $D(s, d, b, q)$ is blocked. The algorithm performs a simple local search through all the connections identified by the set C which includes all

```
function findLSPToPreempt (G', C, D) returns LSPSet
1.  S := ∅; P := ∅
2.  for each low-priority LSP_i ∈ C crossing single-hop lightpaths
3.        ν_i := assignWeightSH (LSP_i, b)
4.        S := S ∪ {<LSP_i, ν_i>}
5.  endfor
6.  if (S ≠ ∅) then
7.        <pLSP> ← pickLowest (S, b)
8.        C := C − {pLSP}
9.        P := P ∪ {pLSP}
10.       C := C ∪ {D}
11. else
12.   for each low-priority LSP_i ∈ C crossing multi-hop lightpaths
13.       ν_i := assignWeightMH (LSP_i, b)
14.       S := S ∪ {<LSP_i, ν_i>}
15.   endfor
16.   if (S ≠ ∅) then
17.       pLSP ← pickLowest (S, b)
18.       C := C − {pLSP}
19.       P := P ∪ {pLSP}
20.       C := C ∪ {D}
21. return P
```

Figure 6.3: The local preemption algorithm (LPA)

the LSPs originated or crossing node s, with final or intermediate destination d, looking for the best low-priority LSP (or LSPs) to preempt in order to leave its route to the incoming request D. When looking for one or more LSPs to preempt in the network, LPA first searches for LSPs carried over some direct single-hop lightpath from s to d (lines 2–10), and then for LSPs carried over a multi-hop lightpath (lines 12–20). In fact, due to the constraint over the maximum number of o-e-o conversions, it is better to route HP traffic over direct lightpaths, while leaving multi-hop paths to LP traffic. Furthermore, it is worth noticing that LPA would consider the extended graph $\mathcal{G}'$ made of LEs which respect the constraint $h_l \leq H_{max}$.

The basic idea of the algorithm is to assign a weight to all the potential preemptable LSPs, so that the ones with the lowest weight will be preempted.

In Figure 6.3 the preemptable connections will be contained in the set P. The algorithm contains two different functions to assign the weights to the LSPs according to the request bandwidth b of the incoming request. As shown in Figure 6.3, the function is different for the single- and multi-hop case.

In the first case, the function `assignWeightSH` assigns weights to each low-priority LSP i with bandwidth b_i and crossing LE j, in order to preempt first a single LSP with the lowest bandwidth such that the available bandwidth on the LE will be sufficient to accomodate the new request (i.e. $b_i + \rho_j \geq b$ where ρ_j is the residual bandwidth on LE j), then one with bandwidth equal or bigger than b and if both these options fail, two or more low-bandwidth LSPs crossing the same LE. The corresponding weight ν_i is given by the formula:

$$\begin{cases} \nu_i = b_i & (b_i + \rho_j \geq b) \\ \nu_i = max\nu + b_i & (b_i + \rho_j < b) \end{cases} \tag{6.1}$$

where $max\nu$ is the maximum value of ν_i when ($b_i + \rho_j \geq b$).

In the multi-hop case, the function `assignWeightMH` considers only low-priority LSPs crossing two or more LEs along their route. Due to the constraints on the maximum number of conversions for high-priority requests, here weights are assigned in order to preempt first LSPs crossing the lower number of o-e-o conversions. Furthermore, in order to avoid the splitting of some high-priority request over two or more lightpaths, only a single low-priority LSP would be preempted.

For each LSP i originated or crossing node s and with final or intermediate destination d, we must find first the crossed LE j with the minimum residual bandwidth σ_j. Then, if b_i is the bandwidth used by this connection, the LSP itself is considered a preemptable connection only if $b_i + \sigma_j \geq b$. The function `assignWeightMH` assigns weights to each low-priority LSP i such that an LSP with the minimum number of o-e-o conversions and the lowest bandwidth (but such that $b_i + \sigma_j \geq b$) is given the smallest weight. The corresponding

weight ν_i is given by the formula:

$$\begin{cases} \nu_i = c_i + b_i \cdot C_{max} & (b_i < b) \\ \nu_i = max\nu 1 + c_i & (b_i = b) \\ \nu_i = max\nu 2 + c_i + b_i \cdot C_{max} & (b_i > b) \end{cases} \tag{6.2}$$

where b_i is the LSP bandwidth and c_i the number of experienced conversions, while $max\nu 1$ and $max\nu 2$ are the maximum value of ν_i when ($b_i < b$) and ($b_i = b$) respectively.

In both cases the preemptable LSPs are added to the ordered set S (lines 4 or 14), while the function `pickLowest` takes the best LSP(s) to preempt from S according to the above specified criteria. The final step is to eliminate $pLSP$ from the set of active connections C and then to update this set with the incoming request D whose route would follow the path traversed by $pLSP$. The returned LSP(s) must be rerouted by using some dynamic grooming algorithm, or blocked if necessary. It is worth noticing that no lightpath disruption happens with the described mechanism; in particular, no lightpaths are torn down while new lightpaths are set up only if the preempted connections cannot find a path in the virtual topology when rerouted.

When compared to GPA, LPA allows to minimize the number of LSPs to reroute, which are more than one only in the case we need to preempt some low-bandwidth LSP, with great benefit for the network disruption. Furthermore, by limiting the execution of the preemption algorithm to the ingress router s only, instead of having an algorithm executed in many LSRs along the request path as in GPA, very few RSVP messages would flow through the network to manage the preemption. In particular the signaling involves the ingress router and some edge LSRs only if the low-priority LSPs to reroute are not originated in s.

6.3 Simulation results

To evaluate the performance of the proposed Traffic Engineering scheme, an extensive set of experiments has been executed. Simulations have been performed on two topologies: the medium-sized topology of [63] and a slightly simplified version of the Sprint topology [51], with 11 POP nodes (corresponding to G-OXCs), 19 non-grooming OXCs and 45 fiber links.

Each wavelength has a full capacity $g = 10$ units, and connection requests have bandwidth demand b_i distributed uniformly between 1 and 3 units, independently from their priority. Requests arrive between each ingress-egress pair according to a Poisson process with an average rate λ, and their holding times are exponentially distributed with mean $1/\mu$, while the network load in all the experiments is given by $\rho = \lambda/\mu$ Erlangs. Ingress and egress router pairs for each LSP set-up request are chosen randomly. The network is loaded with 50000 requests during one trial, and the performance are evaluated by considering average values calculated over 10 runs. The percentage of traffic routed in the network is 60% for LP traffic and 40% for HP traffic. The number of wavelengths considered in all the tests is $W = 4$.

The physical constraints used in the experiments for the high-priority traffic depend on the optical network topology. In particular, H_{max} depends on the network diameter, while a small C_{max} can be very restrictive for topologies where few core nodes have LSR capabilities. In the considered experiments, the values $H_{max} = 4$ and $C_{max} = 1$ have been chosen for the medium-sized topology [63] and $H_{max} = 5$ and $C_{max} = 2$ for the Sprint topology. Because the results are consistent, we present mostly the graphs related to the well-known topology of [63], while the graphs related to the Sprint topology are reported only when some specific result must be highlighted.

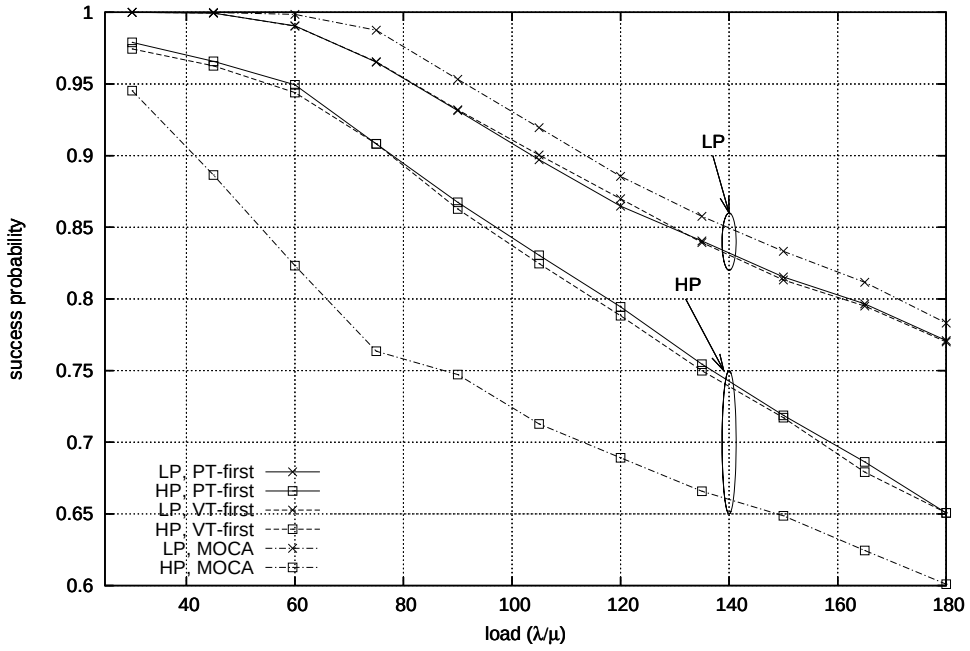


Figure 6.4: Success probability for MOCA vs. `PT-first` and `VT-first`

6.3.1 Comparison between MOCA and proposed grooming algorithms

The first set of tests compare the "QoS-aware" grooming algorithms `VT-first`, `PT-first` and the Minimum Open Capacity Algorithm (MOCA) proposed in [63].

Figure 6.4 shows the corresponding success probability, when traffic requests are limited only to some specific ingress and egress router pairs (MOCA can work only when this strong assumption is considered). The standard deviations are hardly visible (in the order of 10^{-4})) and therefore not shown in all graphs related to success probability. MOCA performs the better for LP traffic, when no requirements are needed to route successfully a connection, while its performance is much worse than our grooming algorithms for HP traffic. This behavior is due to the fact that MOCA is a dynamic grooming algorithm whose main objective is to perform load balancing distribution of the traffic: because the average number of physical hops crossed by each request is very high, it

performs very badly when HP traffic must be routed over the network. Actually, the better performance for LP traffic can be justified also by the fact that more room is left free due to the higher probability of blocking for HP traffic. Furthermore this Figure shows that the resulting blocking probability is very high even for very low network load due to the restrictions on the ingress and egress router pairs.

Another important aspect to consider is the complexity of a dynamic grooming algorithms. In fact when a new LSP needs to be routed over the network with MOCA, the maximum flow between each ingress-egress pair nodes has to be computed in $O(n^2\sqrt{m})$ [63] before applying the traditional $O(nm)$ SPF algorithm. Our grooming algorithms instead route each connection request by using the SPF algorithm only. As an example, one run of 10.000 requests with $W = 8$ and $\lambda/\mu = 200$ took 14 seconds for our algorithms and 13.5 minutes for MOCA on a 1.7 GHz Pentium IV computer running the GNU/Linux operating system.

6.3.2 Comparison between local and global preemption schemes

As expected, for all the proposed dynamic grooming algorithms a higher blocking probability is experienced by high-priority traffic. In the following, we analyze the impact of the preemption mechanism proposed in Sect. 6.2.2. In the rest of this section, we relax the assumption on the position of the ingress-egress router pairs, which are randomly selected every time a new request is loaded in the network.

Figure 6.5 shows the success probability for the `PT-first` dynamic grooming algorithm when a "local" preemption mechanism (LPA) is applied. `VT-first` performs very similarly, thus results are not included: in both cases the obtained gain is quite high. In particular, by using LPA, the success probability for HP traffic increases by about 14%, while it decreases dramatically for LP traffic.

Figure 6.6 shows the performance of LPA when `PT-first` and `VT-first`

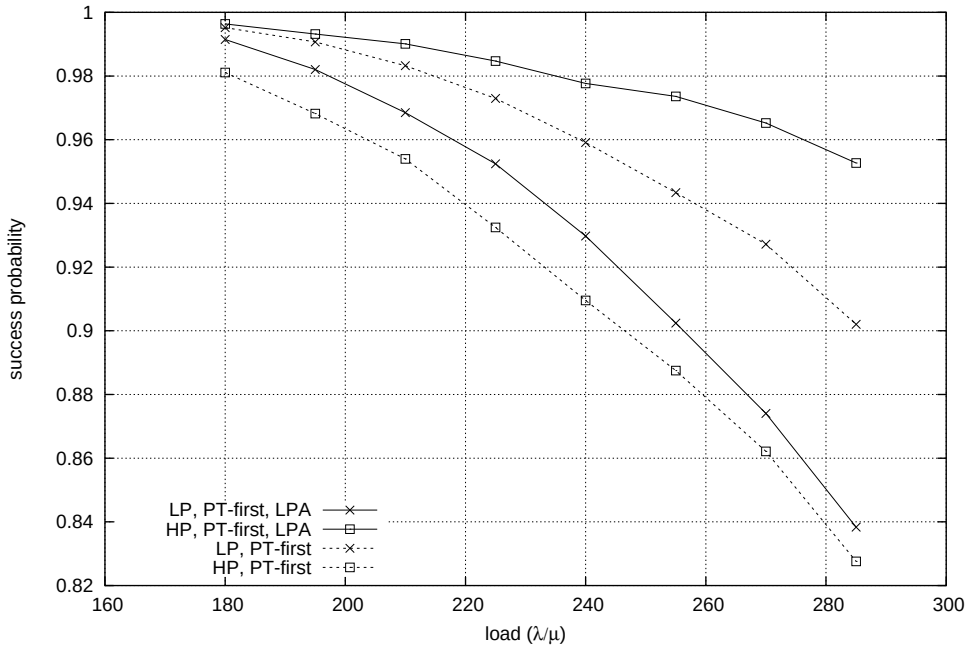


Figure 6.5: Success probability with and without LPA: `PT-first`

is applied. Compared to Figure 6.4, the success probability is increased dramatically for HP traffic to the detriment of LP traffic. It can be noticed that when the `PT-first` grooming algorithm is applied, the success probability for the HP traffic class is always higher than the one obtained with `VT-first`. This can be explained by the implicit mechanism used in `PT-first`, which always tries to set-up a new lightpath (a direct one) when a new request is arrived, thus guaranteeing a higher connectivity in the virtual topology and then more available routes for HP traffic.

Figure 6.7 shows a comparison of the success probability when the proposed LPA and the optimal preemption algorithm (GPA) are applied: only the results when the `PT-first` grooming scheme is applied are shown, because `VT-first` performs very similarly. The GPA mechanism considered in these simulations is implemented by using the mechanism proposed in the $MinConn$ algorithm [87]. LPA performs quite well compared to the optimal algorithm,

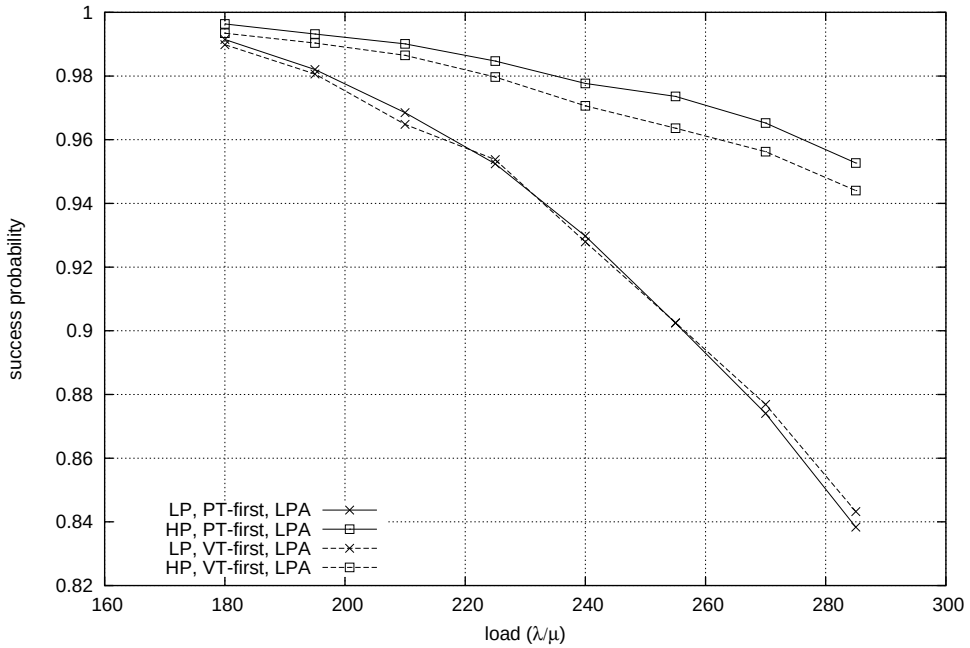


Figure 6.6: Success probability for `PT-first` and `VT-first` with LPA

which always find a route for HP requests. In fact the success probability is high (more than 90%) even at high network loads, when the LP traffic experience a higher blocking probability.

Figure 6.8 shows how the two types of traffic fare when one considers the average number of o-e-o conversions in the absence of service differentiation and when the proposed `LPA` preemption algorithm is adopted, jointly with `VT-first` and `PT-first`. This value gives in fact an estimate of the delay incurred by services carried over the two types of connections. When using `PT-first` the corresponding delay is lower for both classes of service. This can be explained by the implicit mechanism used by this grooming algorithm, which gives preference to direct lightpath when a new request arrives, thus reducing the average number of hops at IP level. Another interesting result is that the proposed preemption mechanism does not impact dramatically on the overall delay for high-priority requests, which is kept to very low values even at

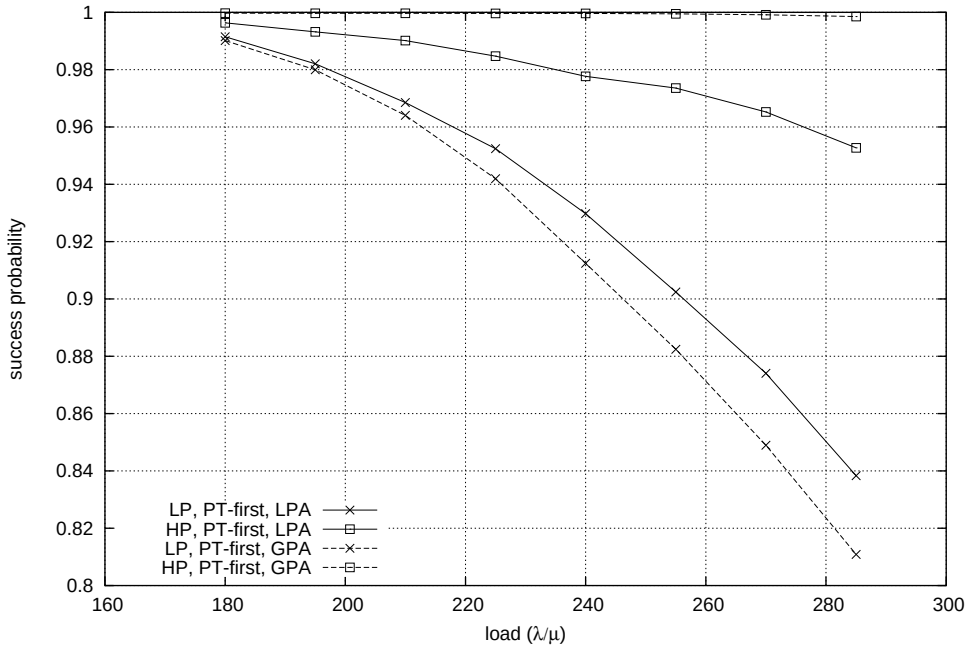


Figure 6.7: Success probability with LPA and GPA: `PT-first`

high network loads.

6.3.3 Impact of the TE scheme on the network disruption

When considering the impact of the proposed TE scheme on the network disruption there are two main parameters to consider: the percentage of rerouted or blocked LP LSPs and the number of lightpaths which are set-up when LP LSPs must be rerouted in the network to leave room for incoming HP connection requests.

Figure 6.9 shows the percentage of rerouted LSPs, calculated as number of rerouted LSPs over the total number of LP LSPs routed with success. As expected, when using the proposed LPA this ratio is much lower than in the optimal case: in particular for the medium-sized topology the rerouting ratio for LPA is roughly the half compared to GPA (upper plot), while for the Sprint topology is less than one fourth (lower plot). The same behavior has been ver-

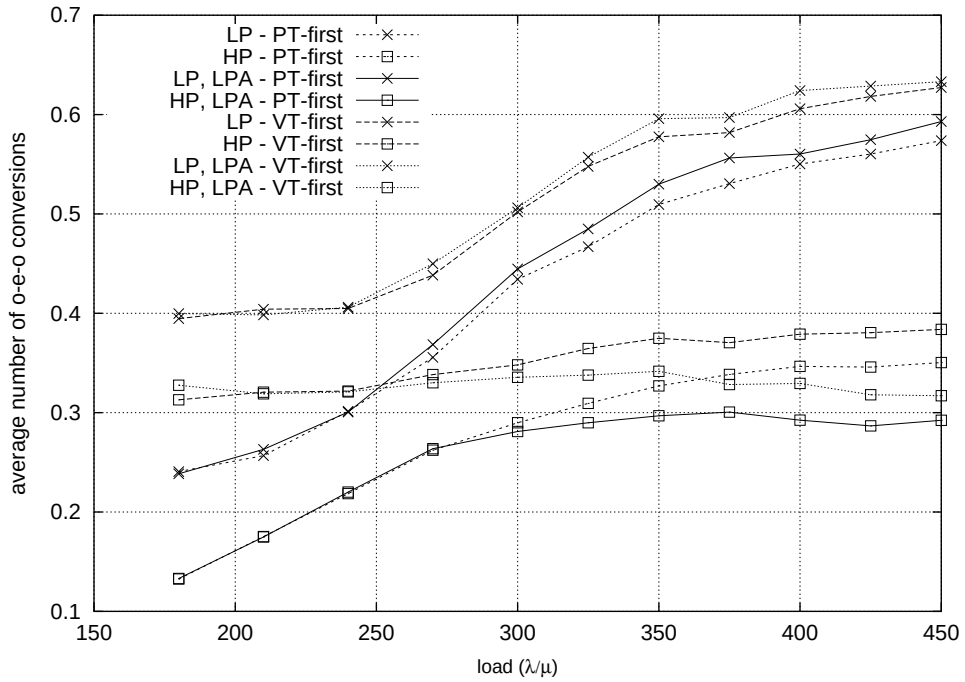


Figure 6.8: Number of o-e-o conversions when using `VT-first` and `PT-first`

ified when the ratio of blocked LP LSPs (preempted connections which could not be rerouted in the network) is considered. In both cases, the lowest ratio of preempted LSPs is obtained by using the `PT-first` grooming policy instead of `VT-first`.

When instead the absolute number of set-up lightpaths when LP LSPs are rerouted is considered (Figure 6.10), GPA sets up between 40% and 60% more lightpaths on average compared to LPA in the medium-sized topology, while for the Sprint topology it sets up 80% and 120% more lightpaths. In both cases the lower number of new lightpaths is reached when `PT-first` is used. In fact, by using `VT-first` the virtual topology would be highly loaded on average, thus forcing the set-up of new-lightpaths when a LP LSP needs to be rerouted.

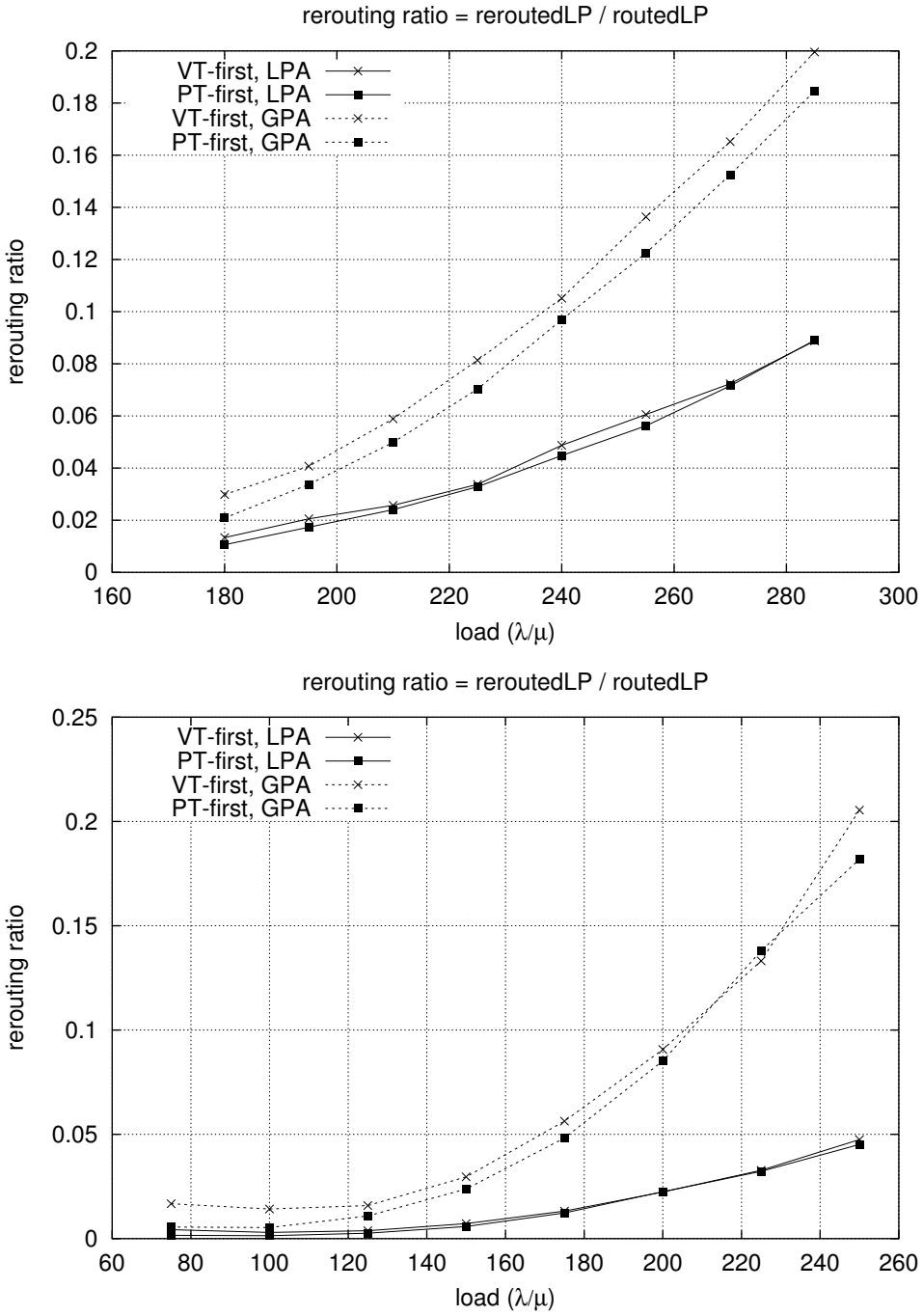


Figure 6.9: Rerouting ratio for medium-sized topology (upper plot) and for Sprint topology (lower plot).

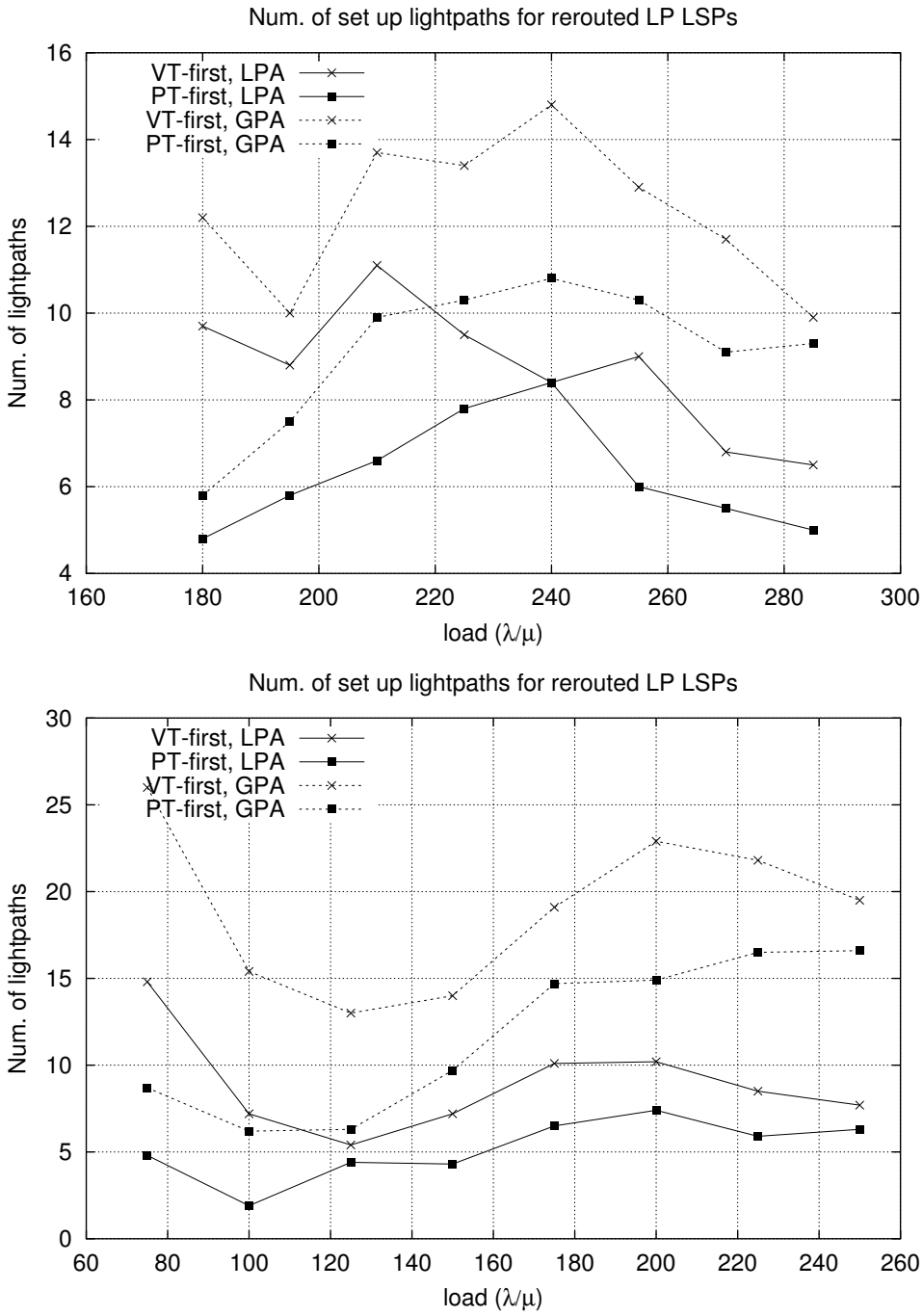


Figure 6.10: Number of set-up lightpaths due to reroutings for medium-sized topology (upper plot) and for Sprint topology (lower plot).

6.4 Conclusions

A novel Traffic Engineering scheme for IP over Optical networks to efficiently route sub-wavelength requests with different QoS requirements has been proposed. Compared to previously proposed TE schemes, the objectives are to minimize the rejection probability for high-priority traffic by respecting specific constraints on the maximum tolerable end-to-end delay and packet-loss ratio at the same time. The proposed scheme consists of an on-line dynamic grooming scheme which routes an incoming request by respecting specific QoS requirements, while a preemption algorithm guarantees that high-priority requests experience a reduced blocking probability when compared to low-priority ones.

Simulations performed on different topologies show that the proposed "local" preemption mechanism LPA minimizes the network disruption both in term of number of preempted LP connections and new set-up lightpaths. The best results, even in term of success probability, are obtained when the `PT-first` grooming policy is applied, i.e. when the set-up of new lightpaths is preferred to the routing of incoming requests over the existing virtual topology. An important aspect of this mechanism is the reduced signaling complexity: in fact, by running the preemption algorithm in the ingress router only, very few edge LSRs are involved in the signaling, thus very few RSVP messages flow through the network to manage the preemption.

Chapter 7

Dynamic grooming in realistic IPO Networks

Traffic grooming is the multiplexing capability aimed at optimizing the capacity utilization in transport systems by means of the combination of low-speed traffic streams onto high-speed (optical) channels. As described in Sect. 3.3, dynamic grooming is basically a routing problem in a multi-layer network architecture (e.g. an IP over Optical network), since the objective is to find the "best" path to route traffic requests arriving dynamically to grooming nodes.

RFC 3717 defines three interconnection models for IPO networks: overlay, augmented and peer. The peer and the augmented models are appealing because sharing the knowledge base between the two layers allows running an integrated routing function, by using, for instance, an auxiliary graph, as done in [126]. The integrated management enables a better usage of the network resources. However, both models seem not feasible in the near term due to the tight integration between the two levels and scalability issues regarding the amount of exchanged information. As described in Sect. 2.3.1, the overlay model is instead technically feasible, since it only requires the definition of an interface between the IP and optical level and dynamic lightpath capabilities in the optical level.

Surprisingly, even if the peer and augmented models do not seem realizable in the near future, most of the dynamic grooming algorithms proposed in the

literature implicitly consider such models (see Sect. 3.3.2 for an overview of them). Only a few dynamic grooming algorithms based on the overlay model have been proposed so far (see Sect. 3.3.1), and the interaction between different routing strategies adopted in the IP and Optical levels was never assessed. However, none of the works on grooming in IPO networks adopted a realistic traffic model. The traffic loading the IPO network is always modelled like a traditional circuit switched traffic, i.e. CBR (Constant Bit Rate) connections characterized by the bit rate and duration. Any realistic evaluation of algorithms to be deployed within the Internet, should instead capture at least the basic characteristics of Internet traffic. As shown in [24], considering the adaptivity of traffic has a deep impact on the network performance and on routing algorithms in particular. The reason lies in the feedback nature of the interaction of elastic traffic with the network: the network status (e.g., congestion) induces a reaction in the source behavior that, depending on the control signal[1] can be a positive or a negative feedback. It is obvious that a positive feedback has, to say the least, a noxious impact on performance, since congestion, or any other performance detrimental status, is exasperated by the positive feedback.

As usual in closed loop systems with delay, the nature of feedback (positive or negative) can change with changing conditions, so that, for instance, a negative feedback at low loads can change to a positive feedback at high loads, leading to instability phenomena.

In this chapter a formal definition of dynamic grooming based on graph theory is defined in a general interconnection model and specialized to the case of the overlay model. A family of grooming policies is proposed in the overlay architecture and the existing dynamic grooming proposals are assessed as a special case of it. For the first time to the best of our knowledge, the performance analysis of these policies is evaluated by considering a Dynamic Statis-

[1] We use the term "*control signal*" though it is not necessary to have a notification protocol to have feedback. Implicit signals, network measures, or simply source-destination interaction can carry the feedback information.

tical Multiplexing (DSM) grooming approach (see Sect. 3.3). In particular, a simple analytical model highlighting the interaction effect between the IP routing and the optical layer is analyzed to assess the impact of traffic elasticity on dynamic grooming. Finally, performance and tradeoffs of different policies are discussed and explained in both regular and irregular topologies, also discussing the impact of adopting TE techniques in the IP or optical level and unfairness issues inherent to overlay dynamic grooming.

7.1 Problem formulation

We present here a formal framework based on graph theory for the definition of dynamic grooming policies.

7.1.1 A formalism for dynamic grooming

IPO networks are based on two layers: the *optical*- and the *data-layer*. The optical-layer is based on OXCs interconnected by fiber links. As described in Sect. 3.3, G-OXCs are the bridge between the optical-layer and data-layer. In the rest of the chapter we are considering a Multi-hop partial or full Grooming OXC architecture for cross-connects. Since a G-OXC is *also* an IP router, the data-layer consists of routers interconnected with a virtual topology made of all the lightpaths which have been set up in the optical-layer.

It is therefore necessary to define a topological graph for each of the layers:

- $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is the physical topology, where $\mathcal{N}$ is the set of vertices v_j (OXCs) and $\mathcal{E}$ is the set of edges e_{jk} (fiber links) connecting vertices v_j and v_k, without loss of generality we compact multiple fibers into a single edge;

- $\mathcal{G}^\nu = (\mathcal{N}^\nu, \mathcal{E}^\nu)$ is the virtual topology, where $\mathcal{N}^\nu$ is the set of vertices v_j^ν (IP routers), $\mathcal{N}^\nu \subseteq \mathcal{N}$, and $\mathcal{E}^\nu$ is the set of edges $e_{jk,q}^\nu$ (virtual links)

connecting vertices v_j^ν and v_k^ν. Each edge in $\mathcal{E}^\nu$ corresponds to a lightpath in the optical-layer. Because there can be more than one lightpath between any two nodes an additional identifier q is required for uniqueness[2].

In the rest of the chapter, the superscript ν is used to specify vertices, edges and paths belonging to the virtual topology. When the superscript is absent, we refer to the physical topology.

Each edge e_{jk} in $\mathcal{G}$ is assigned a vector of properties $\bar{w}_{jk}$ describing any static or dynamic (possibly vectorial) metrics pertaining to physical or traffic-related characteristics of the link. Similarly, a property vector $\bar{w}^\nu_{jk,q}$ is assigned to each edge of $\mathcal{G}^\nu$.

In the optical-layer, a path $\pi_p(v_j, v_k)$, or simply π_p, of length n is defined as a sequence of n distinct edges e_{ih} joining v_j and v_k where $v_j, v_k \in \mathcal{N}$, $e_{ih} \in \mathcal{E}$, $\pi_p(v_j, v_k) = \{e_{ji}, e_{ih}, ..., e_{zk}\}$. The value of p is unique in $\mathcal{G}$ and identifies explicitly the path. This identifier is required since several parallel paths may exist between the nodes v_j and v_k. The path π_p is a *lightpath*, and it corresponds to a specific wavelength if no wavelength conversion is considered[3].

Let $\models$ be the operator that maps a lightpath in the physical topology onto an edge of the virtual topology: $e^\nu_{jk,q} \models \pi_p$ if the path π_p joins the two vertices $v^\nu_j, v^\nu_k \in \mathcal{N}^\nu$. In the data-layer, a path $\pi^\nu_t = \pi^\nu_t(v^\nu_s, v^\nu_d)$ is a sequence of n distinct edges $e^\nu_{ih,q} \in \mathcal{E}^\nu$, t is a unique identifier to distinguish multiple parallel paths.

Fig. 7.1 presents an example. The maximum number of wavelengths per link is $W = 2$. Let's assume that the following five paths have been set up in the optical-layer, where the superscript (i) is used here to specify the corresponding wavelength[4]:

[2]The virtual topology varies in time as lightpaths are set up and torn down. Notice also that $\mathcal{G}^\nu$ has nothing to do with the *auxiliary graphs* defined in [63, 126, 64], which are abstract representations of both levels assuming complete sharing of the two control planes.

[3]In the rest of the description we will keep this assumption, but extensions to the general case of wavelength conversion capability is trivial.

[4]This notation is used here only to help the reader, and it won't be used anymore in the following, since this

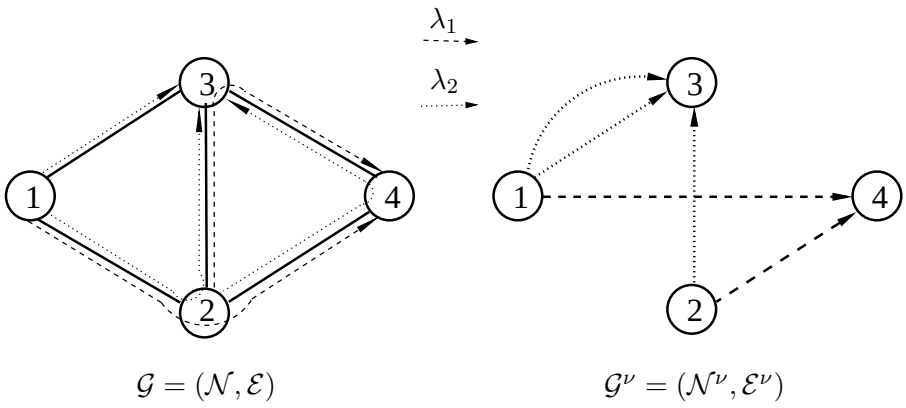


Figure 7.1: Physical and virtual topology for a small 4 nodes network

$$\pi_1^{(1)} = \{e_{12}, e_{24}\},\ \pi_2^{(1)} = \{e_{23}, e_{34}\},$$
$$\pi_3^{(2)} = \{e_{12}, e_{23}\},\ \pi_4^{(2)} = \{e_{24}, e_{34}\},\ \pi_5^{(2)} = \{e_{13}\}.$$

The corresponding set of edges in the data-layer is:

$$\mathcal{E}^\nu = \{e^\nu_{14,1}, e^\nu_{24,2}, e^\nu_{13,3}, e^\nu_{23,4}, e^\nu_{13,5}\},$$

where

$$e^\nu_{14,1} \models \pi_1^{(1)},\ e^\nu_{24,2} \models \pi_2^{(1)},\ e^\nu_{13,3} \models \pi_3^{(2)},\ e^\nu_{23,4} \models \pi_4^{(2)},\ e^\nu_{13,5} \models \pi_5^{(2)}.$$

The identifiers q in edges and p in paths are conceptually different as they are defined in different logical planes, but, as seen in the example, they can assume the same value for simplicity.

For each pair v_j, v_k, the set $\mathcal{P}_\mathcal{G}^{jk}$ is defined as the set of all paths existing between v_j and v_k: $\mathcal{P}_\mathcal{G}^{jk} = \{\pi_p(v_j, v_k) \mid v_j, v_k \in \mathcal{N}, j \neq k\}$. Similarly, a set $\mathcal{P}_{\mathcal{G}^\nu}^{sd}$ is defined as the set of all paths existing between v_s^ν and v_d^ν: $\mathcal{P}_{\mathcal{G}^\nu}^{sd} = \{\pi_t^\nu(v_s^\nu, v_d^\nu) \mid v_s^\nu, v_d^\nu \in \mathcal{N}^\nu, s \neq d\}$.

Given some Routing and Wavelength Assignment (RWA) algorithm $\mathbf{\Lambda}$ in use in the optical-layer, a cost $c^{\mathbf{\Lambda}}(\pi_p)$ is assigned to a path π_p by using a combination of the properties $\bar{w}_{ih}$ of its links $e_{ih} \in \mathcal{E}$. Given $\mathcal{P}_\mathcal{G}^{jk}$ the algorithm $\mathbf{\Lambda}$ selects the

information can be embedded in the path characteristics.

minimum cost path available in the set: $\dot{\pi}_{jk} = \mathbf{\Lambda}(\mathcal{P}_{\mathcal{G}}^{jk})$ that we assume to be unique, possibly by breaking ties with random choices. If no path is available $\mathbf{\Lambda}$ returns $\emptyset$ identifying the empty path.

Let us illustrate the path cost assignment and routing mechanism by using the adaptive routing FPLC (Fixed-Paths Least-Congestion) presented in [68]. A contiguous wavelength is one concurrently available on each link of the path. The cost $c^{\mathbf{\Lambda}}(\pi_p)$ is calculated as the number of contiguous wavelengths on the path, breaking ties with the number of hops in the path:

$$c^{\mathrm{FPLC}}(\pi_p) = u(\pi_p) + \sum_{e_{ih} \in \pi_p} \frac{1}{|\mathcal{N}| + 1} \tag{7.1}$$

where $u(\pi_p)$ is a function counting the contiguous wavelengths currently available on the whole path π_p and $|\mathcal{N}|$ is the number of vertices in $\mathcal{G}$ and it provides a smaller-than-unit weight to break ties based on the number of hops.

In the virtual topology $\mathcal{G}^{\nu}$ we can operate in the same way. Given an IP-based routing algorithm $\mathbf{\Omega}$, the minimum cost path between any two nodes is selected as $\dot{\pi}_{sd}^{\nu} = \mathbf{\Omega}(\mathcal{P}_{\mathcal{G}^{\nu}}^{sd})$.

As an example, the cost function used for the *Minimum Distance* (MD) routing presented in [72] is:

$$c^{\mathrm{MD}}(\pi_t^{\nu}) = \sum_{e_{ih,q}^{\nu} \in \pi_t^{\nu}} \frac{1}{B(e_{ih,q}^{\nu})} \tag{7.2}$$

where $B(e_{ih,q}^{\nu})$ is the bandwidth that is available on the lightpath $e_{ih,q}^{\nu}$ for the request.

A last important set of paths is the set $\mathcal{P}_{pre}$ of pre-established lightpaths in $\mathcal{G}$. $\mathcal{P}_{pre}$ contains a set of permanent lightpaths which guarantees the connectivity of the virtual topology regardless of the traffic pattern and lightpath establishment dynamics. The corresponding set of edges in the virtual topology is called $\mathcal{E}_{pre}^{\nu}$. The set $\mathcal{E}_{pre}^{\nu}$, if present, is always taken into account in the data-layer π_t^{ν} paths computation.

We can therefore define a generic *grooming policy* as a procedure

$$\boldsymbol{G}\left(\boldsymbol{\Lambda}(\mathcal{P}_{\mathcal{G}}^{jk}), \boldsymbol{\Omega}(\mathcal{P}_{\mathcal{G}^{\nu}}^{sd}), \Delta\right) \tag{7.3}$$

where Δ is a set of criteria defining the interaction between the optical and IP level, determining the collaboration between $\boldsymbol{\Lambda}$ and $\boldsymbol{\Omega}$, and defining how many times $\boldsymbol{\Lambda}$ is invoked to setup multiple lightpaths. For instance Δ defines v_j and v_k for the lightpath setup, which can be different from the source-destination pair v_s^{ν},v_d^{ν} of the incoming flow f_{sd}. If present, admission control criteria can be integrated in Δ, which is normally expanded as a set of **if-then-else** clauses.

The set of criteria Δ is influenced by the integration level of the IP and optical control planes (Overlay, Peer or Augmented architectures) and the information base, e.g., status of the optical resources, or cost of the different IP level connections, it operates upon is again function of the control planes integration. For example, in the overlay interconnection model, a reasonable assumption is that $j = s$ and $k = d$, i.e., a single lightpath can be established for each request f_{sd}, and only between the entry and exit nodes of the flow.

So far, we have discussed only the acceptance of new flows, but the grooming policy also defines how to release optical resources when lightpaths are unused. Releasing a lightpath between v_j and v_k means recomputing the set $\mathcal{P}_{\mathcal{G}}^{jk}$ in the optical-layer and consequently deleting an edge e_{jk}^{ν} from the virtual topology and recomputing all the sets $\mathcal{P}_{\mathcal{G}^{\nu}}^{sd}$ that included e_{jk}^{ν}.

7.1.2 Detailing G for overlay architectures

In IPO networks based on the overlay architecture the control planes of the optical and IP levels are separated. Each time an incoming request f_{sd} needs to be routed, there are only two possible options: (i) route it over the current virtual topology $\mathcal{G}_{\nu}$ invoking $\boldsymbol{\Omega}$ or (ii) set-up a new lightpath $e_{sd,q}^{\nu} \models \dot{\pi}_{sd}$ invoking $\boldsymbol{\Lambda}$ and route the request over the new virtual topology $\mathcal{G}_{\nu} \bigcup e_{sd,q}^{\nu}$, invoking $\boldsymbol{\Omega}$ in a second phase.

0. request f_{sd} arrives
1. **if** *exploit optical resources*
2. $\quad e^{\nu}_{sd,q} \models \dot{\pi}_{sd} = \mathbf{\Lambda}(\mathcal{P}^{sd}_{\mathcal{G}})$
3. $\quad \dot{\pi}^{\nu}_{sd} = \mathbf{\Omega}(\mathcal{P}^{sd}_{\mathcal{G}^{\nu} \bigcup e^{\nu}_{sd,q}})$
4. **else** *exploit virtual resources*
5. $\quad \dot{\pi}^{\nu}_{sd} = \mathbf{\Omega}(\mathcal{P}^{sd}_{\mathcal{G}^{\nu}})$
6. $\quad$ **if** $\dot{\pi}^{\nu}_{sd} = \emptyset$ **or** $\dot{\pi}^{\nu}_{sd}$ *is refused*
7. $\quad\quad e^{\nu}_{sd,q} \models \dot{\pi}_{sd} = \mathbf{\Lambda}(\mathcal{P}^{sd}_{\mathcal{G}})$
8. $\quad\quad \dot{\pi}^{\nu}_{sd} = \mathbf{\Omega}(\mathcal{P}^{sd}_{\mathcal{G}^{\nu} \bigcup e^{\nu}_{sd,q}})$

Figure 7.2: General definition of dynamic grooming policies in overlay IPO networks

Fig. 7.2 specifies the procedure (7.3) for an overlay IPO network, without detailing the criteria set Δ. Notice that policies privileging the use of already established lightpaths can always resort to invoke $\mathbf{\Lambda}$ either because no $\dot{\pi}^{\nu}_{sd}$ was found or because the result of $\mathbf{\Omega}$ is refused for any reason.

7.2 Grooming policies

In the scenario depicted above, independently from the definition of $\mathbf{\Lambda}$ and $\mathbf{\Omega}$, the set of rules Δ must define how and when to invoke $\mathbf{\Lambda}$. Although many criteria can be envisaged, the simplest rule is based on the number of IP hops between s and d. In other words, Δ defines as rule the invocation of $\mathbf{\Lambda}$ only if the path selected by $\mathbf{\Omega}$ has more than K hops. We call this policy *Hop Constrained Grooming* $\boldsymbol{HC}(\cdot)$. Additionally, $\boldsymbol{HC}$ can include rules for refusing a logical path π^{ν}_{t} based on congestion measures. With a realistic elastic model of Internet traffic, the definition of congestion is not trivial, since it cannot refer directly to the amount of resources requested by flows. Bandwidth overbooking is a normal practice and we assume that a new lightpath is opened when accepting the flow f_{sd} on the virtual topology would result in assigning it a bandwidth smaller than some given amount applying max-min sharing. The dynamic grooming policies

function $\boldsymbol{HC}(\boldsymbol{\Lambda}, \boldsymbol{\Omega}, \delta(K, \tau_o), f_{sd})$

1. $\dot{\pi}^{\nu}_{sd} = \boldsymbol{\Omega}(\mathcal{P}^{sd}_{\mathcal{G}^{\nu}})$
2. **if** $\{\dot{\pi}^{\nu}_{sd} = \emptyset\}$**or**$\{b(\dot{\pi}^{\nu}_{sd}) < \tau_o\}$**or**$\{H(\dot{\pi}^{\nu}_{sd}) > K\}$
3. $\quad e^{\nu}_{sd,q} \models \dot{\pi}_{sd} = \boldsymbol{\Lambda}(\mathcal{P}^{sd}_{\mathcal{G}})$
4. $\quad \dot{\pi}^{\nu}_{sd} = \boldsymbol{\Omega}(\mathcal{P}^{sd}_{\mathcal{G}^{\nu} \bigcup e^{\nu}_{sd,q}})$
5. **return** $\dot{\pi}^{\nu}_{sd}$

Figure 7.3: The grooming policy ***HC***

studied in [82] and named "Optical-layer-first" and "IP/MPLS-layer-first" are simply the extreme cases for $K = 0$ and $K = \infty$, and have been studied in the simpler case of bandwidth guaranteed traffic.

In order to fix ideas, let's assume that flow requests arrive to the network with two attributes: a *peak transmission rate* B_M and a *minimum requested rate* b_m. If at any time the bandwidth assigned to flow f_{sd} falls below b_m then the flow will close and counted as a "starved" flow, because the network was not able to guarantee its correct completion. B_M and b_m can be included in some SLA (Service Level Agreement) at the IP/Optical interface (see [46] for initial works on Optical-SLA). We thus define a *starvation probability* and not a blocking probability, since the adaptive and elastic nature of Internet traffic does not allow the easy definition of strict admission procedures, even when requests are not single IP flows but large aggregations.

Besides choosing an appropriate K, a network operator may choose to open a new lightpath when routing f_{sd} in virtual topology means it will receive from the beginning a bandwidth smaller than τ_o, possibly a function of f_{sd}. We call τ_o *optical opening threshold*. Clearly, $\tau_o \geq b_m$.

The above criteria specify a Δ with two parameters, say $\delta(K, \tau_o)$, defining the **if-then-else** rules of the generic grooming policy in Fig. 7.2. This leads to the implementable grooming procedure described in Fig. 7.3, and compactly written as $\boldsymbol{HC}(\boldsymbol{\Lambda}, \boldsymbol{\Omega}, \delta(K, \tau_o), f_{sd})$. In Sect. 7.4.7 we shall briefly discuss the impact of using different TE methods in the optical or IP layer, changing $\boldsymbol{\Lambda}$ or

$\mathbf{\Omega}$.

With reference to Fig. 7.3, $H(\pi_t^\nu)$ returns the number of hops in virtual topology path π_t^ν and

$$b(\pi_t^\nu) = \min_{e_{ih,q}^\nu \in \pi_t^\nu} B(e_{ih,q}^\nu)$$

where $B(e_{ih,q}^\nu)$ is the bandwidth available on link $e_{ih,q}^\nu$.

When a lightpath needs to be closed because it is not carrying traffic, the lightpath release could be delayed for a time-interval called *optical closing time-out* τ_{cl}. When no traffic is carried over some lightpath, it is kept open for this timeout period and gets closed only if its state does not change. If not zero, this timeout can be very useful to avoid excessive oscillations in the logical topology, which are notoriously harmful to IP routing.

7.3 The Simulation Tool and traffic models

The implementation of grooming policies in a packet level simulator such as *ns-2* is not convenient for efficiency reasons. Starting from existing tools in our research groups, a simulator has been developed which is capable of handling the layered topological structure of IPO networks as well as several different $\mathbf{\Lambda}$ and $\mathbf{\Omega}$ functions. For a detailed explanation of the tool, named GANCLES, we refer the interested reader to App. A. Presently GANCLES includes different TE techniques both at the optical level and at the IP level. Currently, the family of grooming policies proposed in Sect. 7.2 are implemented, and there is the possibility of choosing different $\mathcal{P}_{pre}$ sets.

As described in this chapter's introduction, the most important features of present Internet applications from the routing point of view are the capacity to adapt the rate to changing network conditions (elasticity) and the need to transfer a given amount of data (compared to the duration of, for instance, a conversational application). The holding time of a flow becomes a *consequence* of the network conditions and not a property of the flow.

To study the impact of traffic adaptation on dynamic grooming algorithms, two different models of elastic traffic are introduced. Both share the characteristic that a flow f_{sd} arrives to the network with a backlog of data D to transmit and both include some form of elasticity, though very different one another.

The first model, that we name *time-based* (TB), simply models the fact that the more congested is the network, the smaller is the throughput the flows get. The flow duration is determined when the flow arrives to the network, based on its backlog and its minimum requested rate (see Sect. A.3 for more details). This model is very simple and does not grab all the complexity of the closed-loop interaction between the sources and the network. In a more accurate model, named *data-based* (DB), traffic flows share the resources on a virtual topology path following the max-min fairness criterion [18], thus mimicking the ideal behavior of a bundle of TCP connections (see Sect. A.3). The acceptance (or release) of a new flow affects therefore not only all the other flows on the same path, but indeed all the flows in the network, since the max-min fair share is completely recomputed updating the estimated closing time of all the flows in the network.

The DB model is clearly much more accurate, closely mimicking the behavior of an ideal congestion control scheme; however its complexity and computational burden are much larger, specially for high loads. Investigating whether (or under which conditions) the simpler TB model is accurate enough in the context of IP over WDM with dynamic grooming, or if it leads to gross approximations can be very important both for theoretic research and for practitioners.

7.4 Results and discussion

7.4.1 Performance indices

The phenomena involved in routing/grooming elastic traffic are rather complex, and often far from intuitive. The following performance indices blend both

user-perceived performance and network operation costs, thus helping in understanding the global performance of the network.

p_b — Blocking probability. It is the probability that a flow is not accepted, either because of some CAC function decision or (in best-effort traffic) because no connectivity can be found between the source and the destination.

p_s — Starvation probability. It is the probability that a best-effort flow closes during its life because it is not receiving service with acceptable quality. A flow f_{sd} closes and drops the network if its instantaneous throughput falls below b_m.

T — Average normalized throughput of completed flows.

$$T = \frac{\sum_{v_s^\nu \in \mathcal{G}^\nu} \sum_{v_d^\nu \in \mathcal{G}^\nu} \sum_{\forall f_{sd}} T(f_{sd})}{\sum_{v_s^\nu \in \mathcal{G}^\nu} \sum_{v_d^\nu \in \mathcal{G}^\nu} \sum_{\forall f_{sd}} 1}$$

$T(f_{sd})$ is the average normalized throughput of flow f_{sd} provided it was not starved. Notice that in a resource sharing environment T is not the average resource occupation divided by the number of flows.

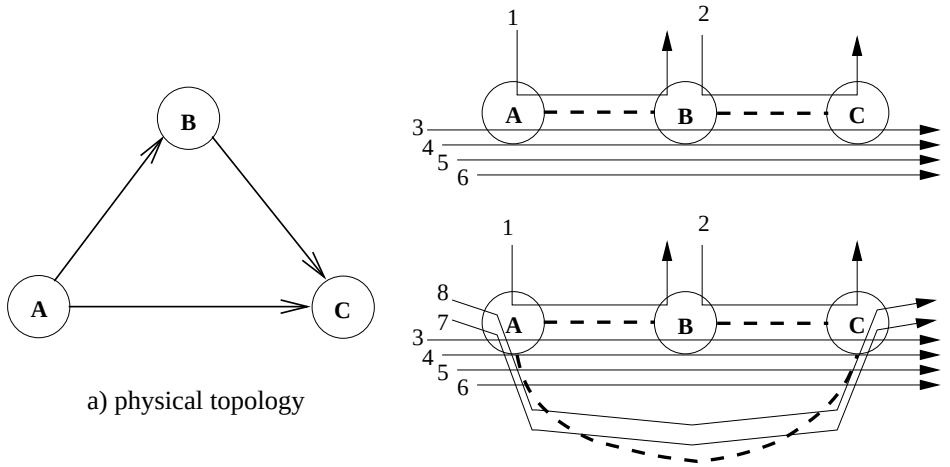
R_c — Routing table change rate. Each time a new lightpath π_p is established or torn down, some of the sets $\mathcal{P}_{\mathcal{G}^\nu}^{sd}$ must be recomputed. The rate of such changes is a good measure of the joint grooming and routing cost within the network.

N_{lo} — Time weighted average number of links per optical path. Each lightpath is weighted by its holding time, so that lightpaths lasting longer are correctly accounted for.

U_d — Distance unfairness index.

$$U_d = \frac{\max_{0<r<|\mathcal{N}|} T^r - \min_{0<r<|\mathcal{N}|} T^r}{T}$$

measures if the resource assignment is fair with respect to physical distance. T^r is the average throughput calculated for node-pairs with hop distance r in the physical topology. It ranges from zero to ∞; any value larger than 1 indicates unacceptable unfairness. This parameter is evaluated only for regular topologies



a) physical topology

b) logical topology with call sequence and routing

Figure 7.4: Simple 3-node topology used for the theoretic verification of results

and uniform traffic, since in other cases it is can be influenced by factors external to the grooming policy.

The goal of a grooming algorithm is maximizing T while minimizing p_s, R_c and U_d.

Before discussing results on regular and irregular topologies, we highlight some peculiar behavior of the grooming policy ***HC*** for $K = \infty$ in a very simple analytical model, that will help in interpreting results in more complex scenarios.

7.4.2 A simple analytical model

Consider the simple 3-node topology of Fig. 7.4 a), where only a single wavelength per link is present, each node in the network has full grooming capability and the active traffic relations are only A–B, B–C, and A–C. Assume that grooming policy ***HC*** with $K = \infty$ is used and, starting with the network empty, the following sequence of flows arrives: AB, BC, AC, AC, AC, AC, ...(AB identifies a flow originating in A with destination B and so on). We

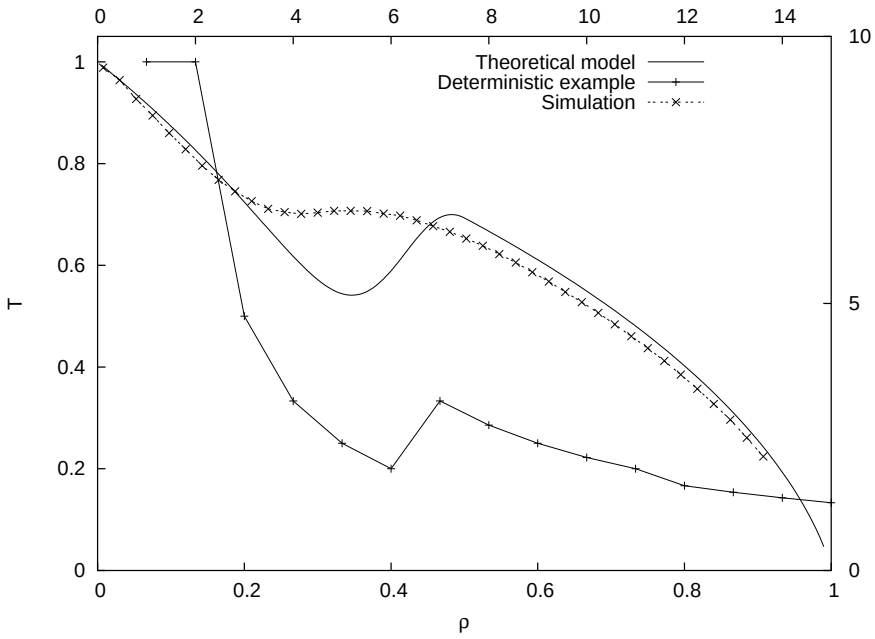


Figure 7.5: Average throughput computed deterministically, via simulation and with a simple stochastic model for the scenario depicted in Fig. 7.4 a)

set $\tau_o = 2$ Gbit/s and $b_m = 0$ Gbit/s and all flows are able to fully exploit the optical path capacity (10 Gbit/s). The average throughput obtained by flows is represented by the solid line with cross marks in Fig. 7.5 (this curve refers to the top x-axis (number of flows) and left y-axis (normalized throughput)), as can be easily seen following the logical topology evolution reported in Fig. 7.4 b).

This example show that with $K = \infty$, it is possible that T increases while the load increases due to the interaction between the IP and optical layer. However, a deterministic example is not enough to draw conclusions. In order to investigate further in the behavior, we have set up a simple (and approximate) queuing model of the same scenario based on processor sharing queues that mimic the max-min resource division. In order to simplify the analysis, we assume that a lightpath is always open on the links A–B and B–C. With this assumption the queuing network depicted in Fig.7.6 represents fairly well the behavior of the network we consider. Assume the three traffic relations offer the same load

ρ (normalized to the optical path capacity) to the network[5]. Queue Q1 represent the lambda on link A–B, queue Q2 represent the link B–C and queue Q3 represents the link A–C. All queues are M/M/1-PS. The routing probability p_{ac} describes the fact that with this grooming policy and lambdas always open on A–B and B–C links, the AC traffic is routed over A–C links only when the τ_o threshold is hit, then it is routed on A–C until this link empties and the relative optical path is closed. $\tau_o = 2$ Gbit/s means trying to open a new lightpath when a new flow would lead some existing lightpath to have more than $N_o = 5$ flows on it.

The exact computation of p_{ac} is complex, because it is in fact due to the superposition of transients and it is not a steady state probability, as the simple model assumes. We approximate it starting from the clients distribution in queues Q1 and Q2 when all traffic is routed through A–B, B–C, and assuming that flows are routed over A–C only when there are more than N_o flows either on A–B or B–C links. The simple model assumes that Q1 and Q2 are independent (which is not true in reality, since flows cross both links, and hence occupies both queues, at the same time), which implies $p_{ac} = 2p_t - p_t^2$ where $p_t = (2\rho)^{N_o}; \rho < 0.5$ or $p_t = 1; \rho \geq 0.5$.

The M/M/1-PS average throughput is computed following the approximate formula derived in [62].

Fig. 7.5 reports, beside the simple deterministic example, results obtained with the simple stochastic model and with simulations (DB traffic model) for $\tau_o = 2$ Gbit/s. The simulation curve does not show the same increase in throughput around the load $\rho = 0.5$ displayed by the model (however, we have observed it for much smaller thresholds). The reason is that the dynamic routing of flows makes the transition from routing the AC traffic mainly through A–B and B–C to routing it mainly over A–C smoother than in the approximate model. In this

[5]The model can cope with different loads on traffic relations; however, the focus here is not on the model capabilities but on the explanation of grooming algorithms behaviors in IPO networks hence we keep the model as simple as possible.

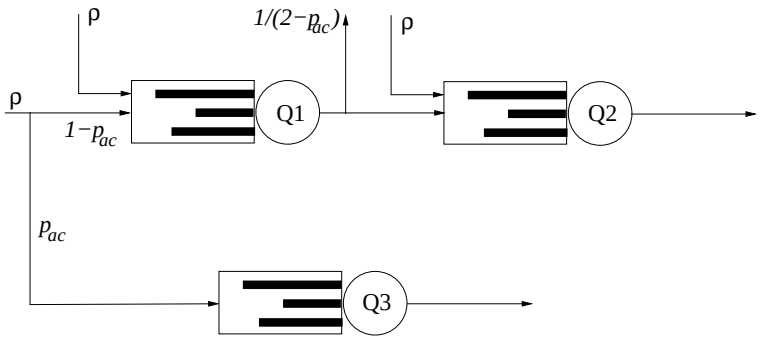


Figure 7.6: Queuing model corresponding to the simple scenario of Fig. 7.4

case we set $b_m = 0$ Gbit/s, so that $p_s = 0$, while other performance indices are not of much interest.

This simple example give some insight on the complex behavior of grooming associated with elastic traffic, which, to the best of our knowledge was never observed in other works, that, using constant-bit-rate like traffic models, cannot observe throughput performance. In the following we study more realistic scenarios (by simulation only), to gain more insight on grooming and elastic traffic interaction.

7.4.3 Networking scenarios

When considering more complex scenarios, a theoretic performance analysis is not feasible due to complexity of the system. We resort to simulations for the performance evaluation, exploring both regular and irregular topologies.

Three different topologies are considered in our study:

- *R8* is an 8 nodes bidirectional ring;
- *MT16* is a 16 nodes mesh-torus network with connectivity four, i.e., each node is connected to the four adjacent nodes in a regular, closed lattice mapped on the surface of a bi-dimensional torus — or a doughnut in pop terms;

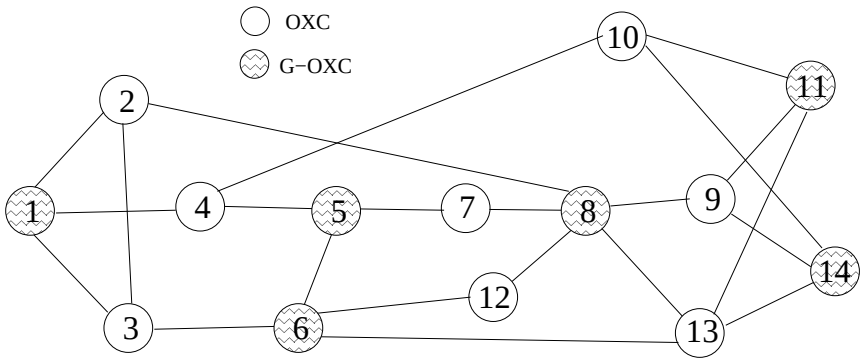


Figure 7.7: Topology *NSF*

- *NSF* is the well-known NSF-net topology with 14 nodes and 21 fiber links. In this topology we assume that only 6 OXCs have grooming capabilities, as shown in Fig. 7.7.

The number of wavelength W is one of the parameters of major interest to investigate the generality and scalability of solutions with respect to the amount of available resources. The considered architecture for G-OXC is Multi-hop full Grooming OXC according to the classification discussed in Sect. 3.3. Each wavelength has a capacity of $g = 20$ Gbit/s. A data-layer traffic source is connected to each G-OXC, generating requests with $B_M = 10$ Gbit/s following a Poisson process. Each flow transfer data whose amount is randomly chosen from an exponential distribution with average 12.5 GBytes. Unless otherwise specified, $b_m = 1$ Gbit/s in all simulations, $\tau_o = 3$ Gbit/s and dynamically opened lightpaths are immediately torn down if they are not used ($\tau_{cl} = 0$). All simulations are run until performance indices reach a 99% confidence level over a $\pm 1\%$ confidence interval around the point estimate. Estimations are carried out with the *batch means* technique. Results are plotted versus the total load L offered to the network.

Unless otherwise stated, $\mathbf{\Lambda}$ is the FPLC algorithm described in Sect. 7.1.1 with first-fit wavelength assignment and $\mathbf{\Omega}$ is the standard fixed shortest path (FSP) algorithm. A uniform traffic pattern is simulated, i.e., when a new flow

request is generated, the source and destination are randomly chosen with the same probability.

7.4.4 Impact of the elastic traffic on grooming policies

The first set of results evaluates the impact of realistic traffic in IPO network by analyzing the behaviour of the two traffic models presented in Sect. 7.3 when considering the two "extreme" grooming policies ***HC*** for $K = 0$ and $K = \infty$. Simulations have been performed on different network topologies, but thanks to the consistency of the results, only the graphs for the *NSF* topology with $W = 4$ are shown. The results reported in this subsection are obtained by considering $\tau_o = b_m = 1$ Gbit/s in all simulations.

Fig. 7.8 (upper plot) presents a comparison of the average normalized throughput T obtained modelling best-effort traffic relations using the TB approach (dotted lines) and the DB approach (solid lines) when ***HC*** grooming algorithm for $K = \infty$ (round marks) and $K = 0$ (cross marks) are used. With the same graphic rules, Fig. 7.8 (lower plot) reports the starvation probability.

The difference in performance results of the two approaches is striking. Let's consider first the grooming policy with $K = 0$. Both approaches show maximum T when the offered load is low; however, they immediately diverge as the offered load increases. Indeed, the DB traffic model shows much faster decrease in T as soon as the offered load increases and this is due to the spreading of congestion over time with a sort of snow-ball effect. On the contrary, the TB traffic model shows a smoother decrease of the average bandwidth.

Analyzing the starvation probability in Fig. 7.8 (lower plot) adds more insight. When the traffic is very low (below 350 Gbit/s) both traffic models show the same, very strange behavior: the starvation increases and then decreases sharply. This form of blocking is independent of the traffic model and it is due to a very aggressive and dynamic use of optical resources that sometimes leads to have no connectivity at the IP level, i.e., a flow request arrives and there is no

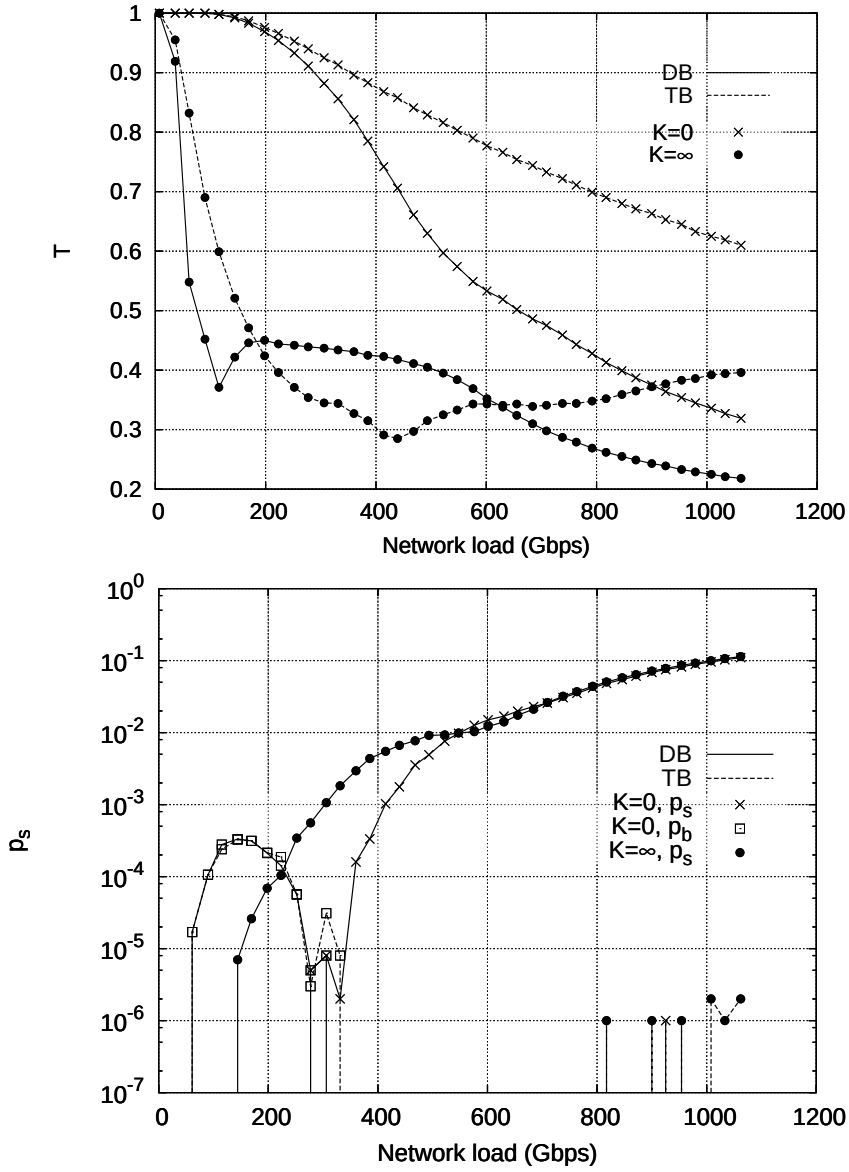


Figure 7.8: Per-flow average normalized throughput T (upper plot) and starvation probability p_s (lower plot) for the DB and TB traffic models, *NSF* topology with $W = 4$

possible path, neither optical, nor through multiple IP hop, between the source and the destination. When the load increases, however, lightpaths become more stable (because there is always traffic keeping lightpaths open) and the probability that the virtual topology is not completely connected becomes negligible. To highlight the difference of this phenomenon from the real starvation, in Fig. 7.8 (lower plot) the curves relative to it are plotted with square marks and indicated as blocking probability p_b. When the load increases further, the two traffic models behavior diverge: the TB model shows no starvation at all, apart from points at very high loads, which shows a blocking probability around 10^{-6}, while the DB model shows a starvation probability increasing steadily. This difference in the starvation behavior enhances the differences in T, since aborting flows causes a waste of bandwidth.

When considering the grooming policy with $K = \infty$ instead, the behaviour of both traffic model is different from the previous one. Both DB and TB T decrease sharply even when the offered load is low, due to the conservative behaviour of this policy. In fact, with $K = \infty$ the minimum number of lightpaths is set up in order to guarantee the minimum network connectivity, and keeps this configuration unchanged until some flow crosses the starvation threshold b_m. Only in this case resources at IP level are increased by setting up new lightpaths. In particular, the T for DB traffic relations decreases very rapidly, causing an earlier set-up of new lightpaths compared to TB traffic. This lead the DB traffic throughput T to "bounce" taking advantage of the higher number of lightpaths in the network, at a load much smaller than for the TB model, that starts increasing again at higher loads. Obviously, both models would show another (and definitive) decrease in T for higher loads, not simulated here. It is interesting to notice that the starvation rate (see Fig. 7.8 (lower plot)) for the TB model is in this case always zero (apart from a single point around load 900 Gbit/s), while the starvation rate of the DB model increases steadily and shows a behavior similar to the DB model in the $K = 0$ case.

From this first set of results we argue that using a more sophisticated, *data-based* traffic model exhibiting a realistic behavior is essential to evaluate the performance of any dynamic grooming algorithm. In the rest of the simulations we therefore consider only the DB traffic model.

7.4.5 Grooming policies behavior on different topologies

The second set of results shows how grooming policy ***HC*** behaves on both regular and irregular topologies when varying K and W.

When considering regular topologies with all G-OXCs and uniform traffic, the load can be expressed also as the relative traffic ρ offered to each network node normalized to the total capacity of its egress links. Given the total number of nodes $|\mathcal{N}|$ in the network, the connectivity degree D, the number of wavelengths per fiber W and their data-rate g, we have: $\rho = \frac{L}{|\mathcal{N}|DWg}$; ρ cannot be defined if the topology is not regular or some nodes do not have grooming capabilities, but it offers a means of comparison between different topologies and different values of W.

In the previous section we observed that dynamic grooming policies using aggressively the optical resources may, from time to time, and specially at low/medium loads, build logical topologies that are not completely connected, a situations clearly unacceptable in operations. Including a pre-defined spanning tree or any other topology ensuring connectivity in the data-layer avoids this problem. As it was mentioned in Sect. 7.1.1, the virtual topology connectivity can be guaranteed with pre-established lightpaths that are never closed. We consider two possibilities to populate $\mathcal{P}_{pre}$:

- A Minimum Spanning Tree (MST) of lightpaths, which connects all the G-OXCs by using the minimal amount of optical resources (for implementation issues, we refer the reader to [5]);
- A 'Physical-Topology' (PT) of lightpaths, which sets up the lowest order

wavelength among each pair of adjacent nodes[6]. Notice that the authors in [129] observed that using a solution similar to PT improves the blocking probability in a context with bandwidth-guaranteed traffic.

In the rest of the chapter, the solution PT is applied for the regular topologies *MT16* and *R8* and MST for *NSF*, since MST in rings and mesh-toruses has poor performance.

We first consider the impact of W on *R8*. Fig. 7.9 and Fig. 7.10 show the impact of using different values of K on T and p_s for $W = 4$ and $W = 8$ respectively. On the bottom x-axis we use L and on the top x-axis we use ρ.

In both cases $K = 1$ ensures the best performance. The performance spread increases with W; results for $W = 12$ confirm this result. This behavior comes from the aggressive use of optical-layer resources with $K = 0$. Setting up lightpaths even when not needed, the optical-layer becomes overcrowded with lightpaths, which leads to blocking lightpath set-up requests when congestion is impending, resulting in a poorly connected virtual topology, reduced throughput T and increased starvation p_s. Increasing K above 1, the performance tends to be similar to $K = \infty$. This is due to the small average distance between nodes, but it also confirms that the best way to use optical resources is trying to build a fully connected mesh in the virtual topology. Although difficult to formally prove and even to show, we have observed that by using $K = 1$ the grooming algorithm tends to build a full-mesh virtual topology when W provides enough resources, and this independently from the physical topology. In our opinion, this is the main reason why setting $K = 1$ guarantees better performance.

Fig. 7.11 shows T and p_s for a *MT16* topology with $W = 8$. The behavior is similar to the one observed in *R8* confirming that the relative merit of grooming policies is not related to the topology.

The analysis of the behavior in different network topologies is completed

[6]This approach is meaningful only if all the network nodes are G-OXCs, because lightpaths between non-grooming OXCs would be unused.

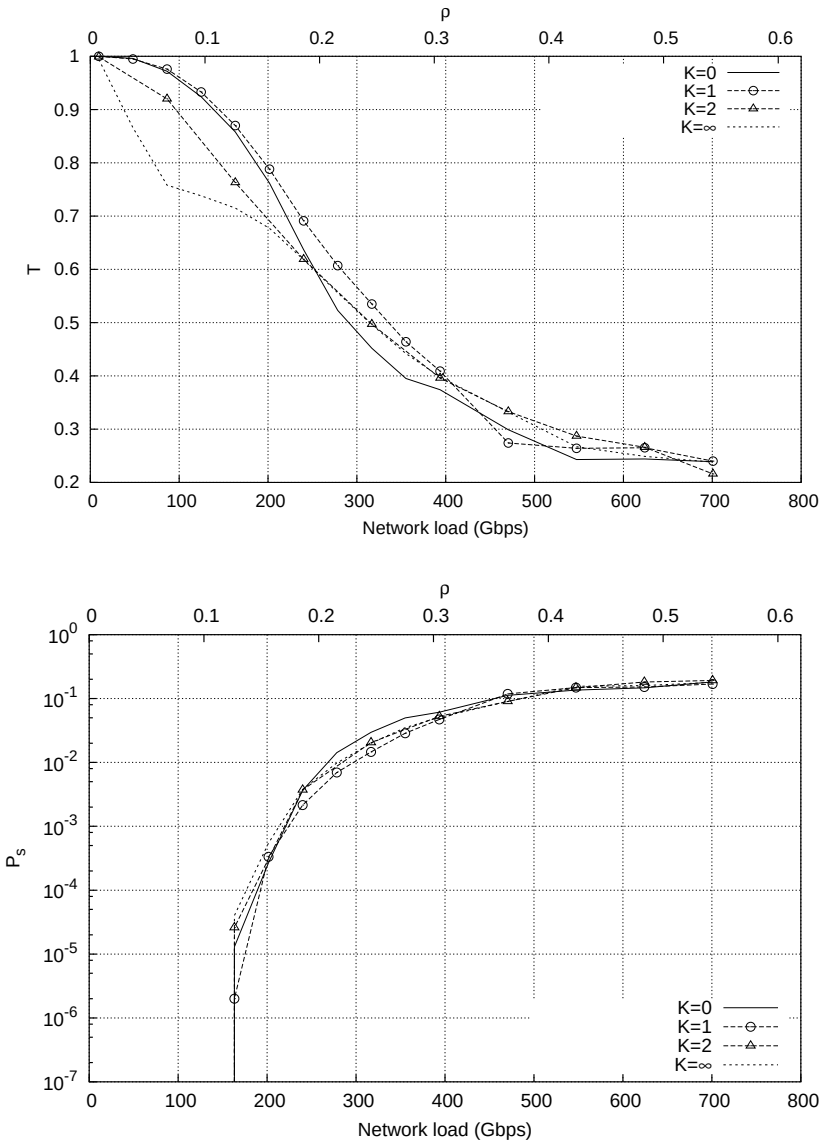


Figure 7.9: Per-flow average normalized throughput T (upper plot) and starvation probability p_s (lower plot) for *R8* with $W = 4$

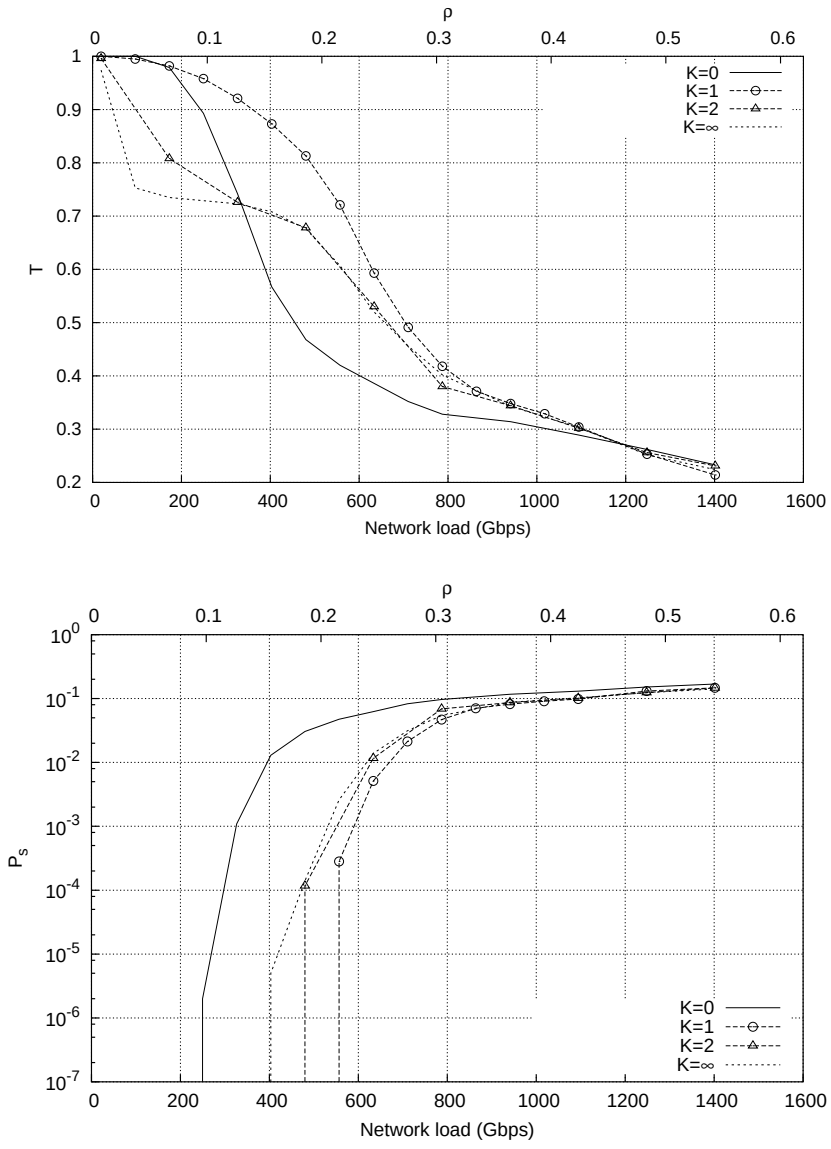


Figure 7.10: Per-flow average normalized throughput T (upper plot) and starvation probability p_s (lower plot) for *R8* with $W = 8$

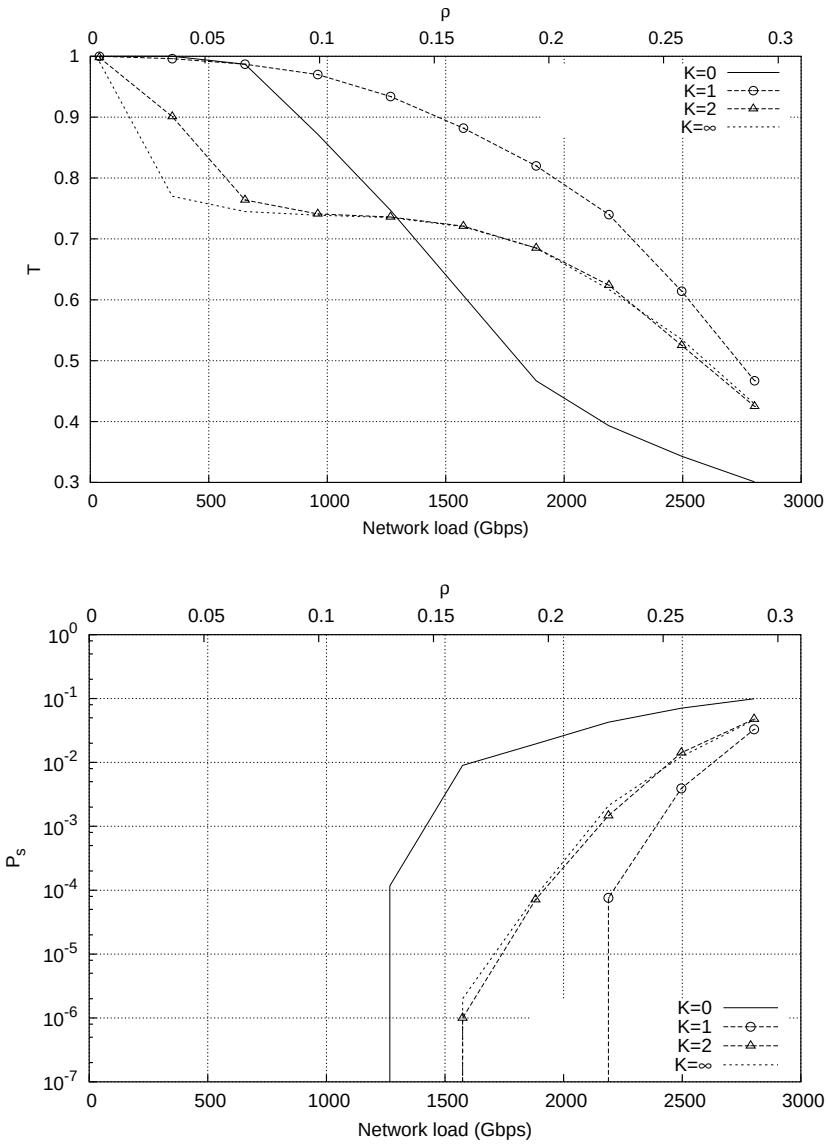


Figure 7.11: Per-flow average normalized throughput T (upper plot) and starvation probability p_s (lower plot) for *MT16* with $W = 8$

with the *NSF*. Fig. 7.12 shows T and p_s when $W = 4$. In this case $K = 0$ shows a higher throughput, but this is due to the higher p_s and not to a better performance: the lower number of flows coexisting in the network increases the throughput for the surviving ones. $K = 1$ ensures the best combination of T and p_s.

Restricting the analysis to $K = 0, 1$ and regular topologies, we analyze the effect of W on performance with the normalized load ρ. Fig. 7.13 compare the behavior on *R8* with $W = 4, 8, 12$. The figure clearly shows that increasing W the choice of $K = 1$ is indeed the best one, keeping the efficiency basically constant as the amount of resources and the traffic increases, while for $K = 0$ the performance decreases drastically.

Fig. 7.14 presents a similar set of results for *MT16* and $W = 2, 4, 8$. These results confirm the conclusions above[7]. Most interesting is that for low W the performances are similar and in both topologies tend to diverge as W increase.

So far we have considered user-level performance. An important cost factor to evaluate grooming policies is how frequently the routing tables of virtual topology need to be updated due to changes induced by opened and closed lightpaths, which is an important cost factor in an IPO network [90]. Fig. 7.15 shows the impact of K on R_c, for both *R8* and *NSF* (similar results have been obtained on *MT16*). In both cases, an aggressive use of the optical resources ($K = 0$) leads to a very costly change rate, while instead more "conservative" grooming policies ($K > 0$) allow lower rates, thus guaranteeing a much safer stability in the data-layer. It is interesting to notice that, after rising quickly with the load, R_c converges to very low values, independently from the topology or from K, as the load increases and the virtual topology becomes stable (lightpaths are closed with very low probability). Allowing a lightpath to be idle for some timeout period waiting for possible new flows before closing it, would obviously reduce

[7]We use smaller Ws on *MT16* because the high connectivity degree associated with a large number of nodes and large W makes simulations extremely CPU and memory demanding.

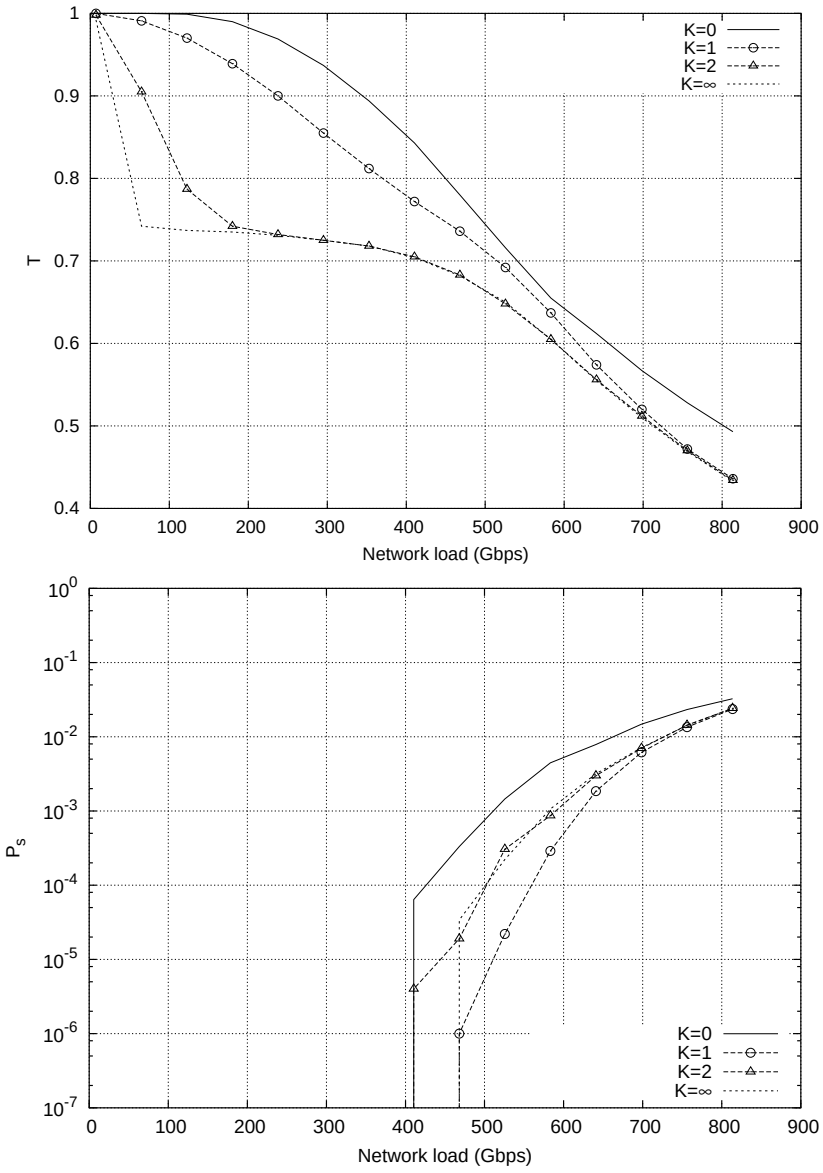


Figure 7.12: Per-flow average normalized throughput T (upper plot) and starvation probability p_s (lower plot) for *NSF* topology with $W = 4$

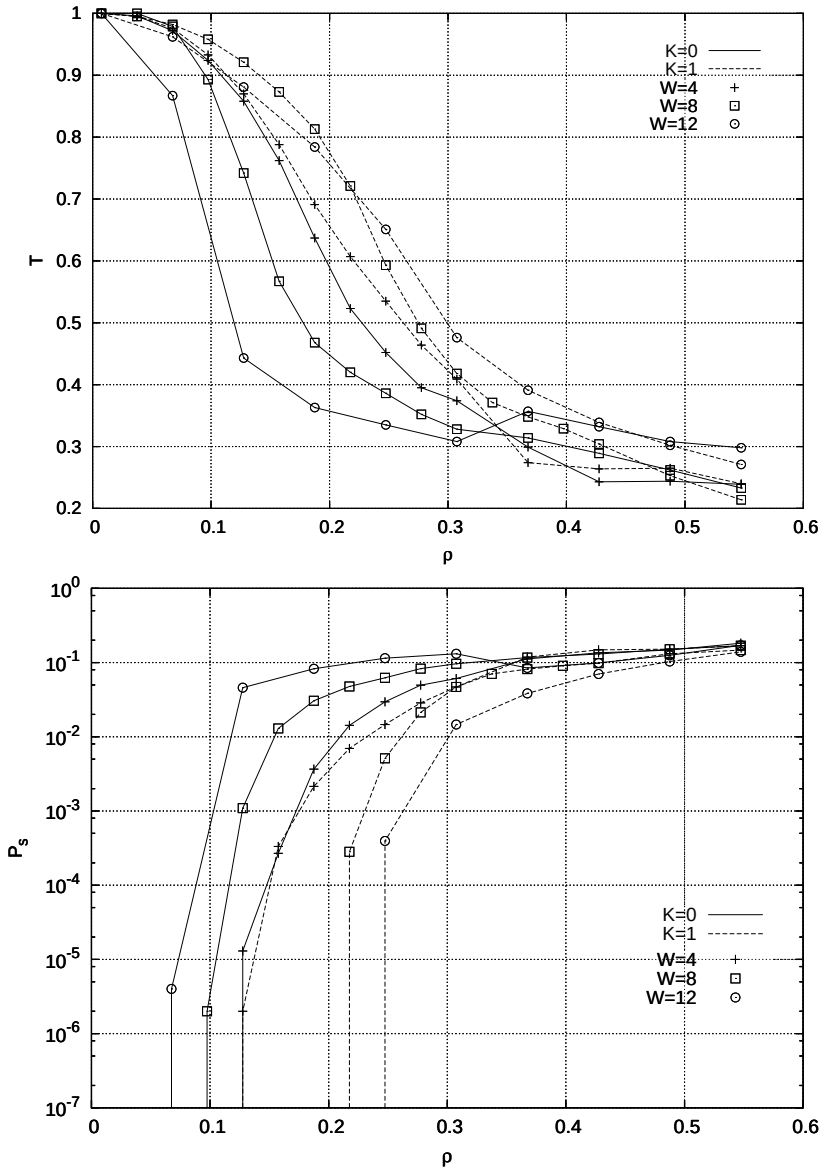


Figure 7.13: Per-flow average normalized throughput T (upper plot) and starvation probability p_s (lower plot) for *R8* with varying W and $K = 0, 1$

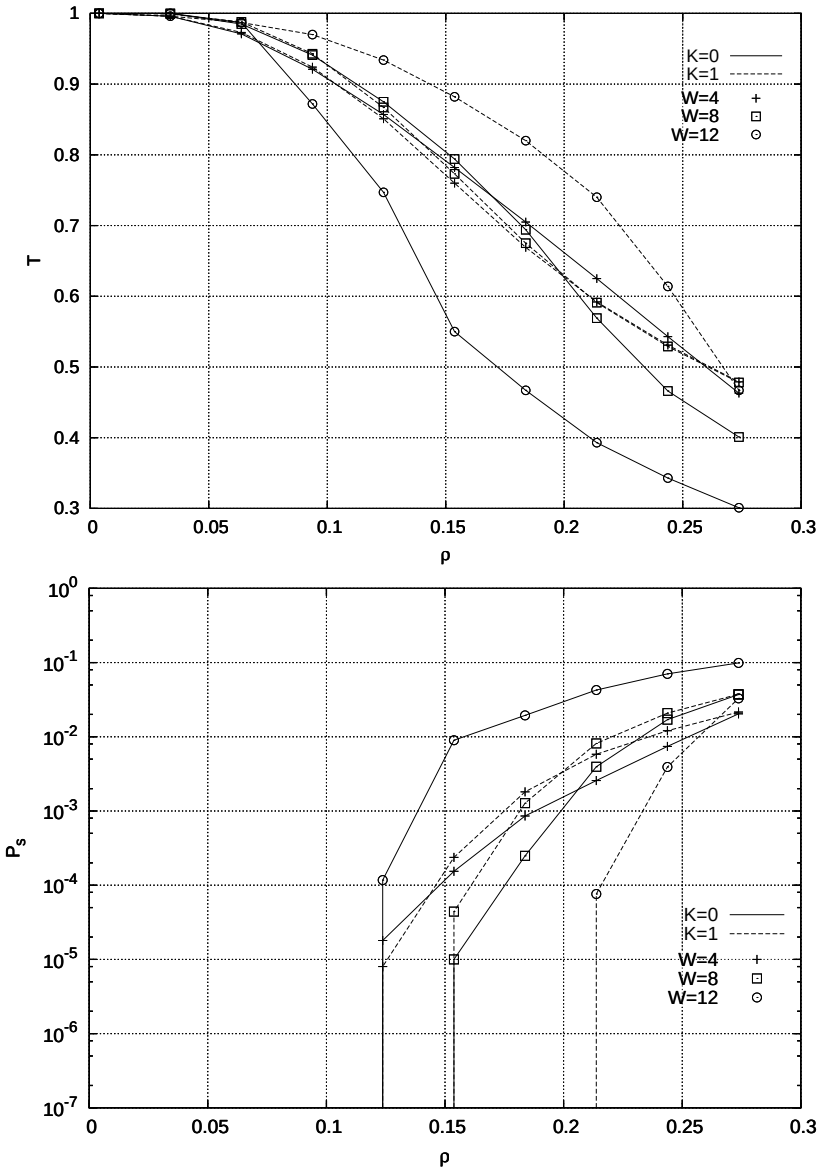


Figure 7.14: Per-flow average normalized throughput T (upper plot) and starvation probability p_s (lower plot) for *MT16* with varying W and $K = 0, 1$

this cost, but also avoid the use of the wavelengths for other traffic.

7.4.6 Fairness issues with elastic traffic

We now concentrate on fairness issues, analyzing how different grooming algorithms in overlay IPO networks influence fairness. In particular we investigate two main aspects that influence fairness: the minimum bandwidth b_m required by incoming flow requests, and the physical distance (in terms of the number of crossed OXCs) between the end-nodes of a flow. This latter problem was addressed in [32] in the context of peer IPO networks with a bandwidth-guaranteed traffic model. Resource sharing in general exacerbate the unfairness, because longer flows share the resources with more flows with respect to short ones, thus they remain longer within the network and the more they remain the more the flows they compete with.

A first set of results regarding the fairness of different grooming algorithms for clients with different SLA requirements is shown in Fig. 7.16 by considering *R8* with $W = 8$. These results have been obtained differentiating the users in terms of the minimum bandwidth b_m they require to the network to avoid starvation. In particular, we consider two classes of users. Class C_1 with minimum bandwidth request $b_{m1} = 2$ Gbit/s and class C_2 with $b_{m2} = 1$ Gbit/s. The hop constraint K is fixed to 0, 1, ∞.

The upper plot shows the average throughput and the lower one the starvation probability. It is clear that the relative merits of grooming policies are unchanged by the presence of classes, but the main result to highlight here is that when considering a realistic traffic scenario, none of the grooming algorithms allows to improve the fairness with respect to the minimum requested bandwidth. The explanation is trivial, but the solution seems to be far more complex, like introducing some form of proportional scheduling in nodes or some form of CAC. In fact, when the traffic shares resources with a max-min fairness criterion, the available network resources are fully shared among all the

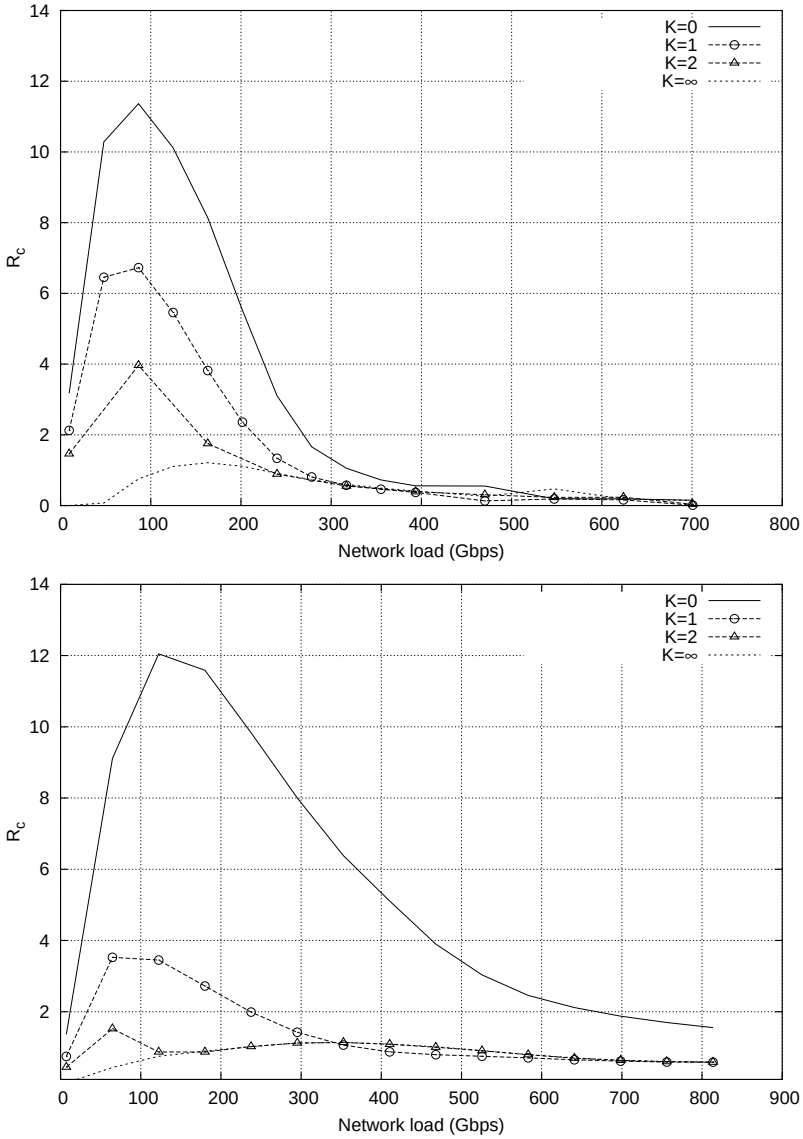


Figure 7.15: Routing table change rate for *R8* with $W = 4$ (upper plot) and for *NSF* topology with $W = 4$ (lower plot)

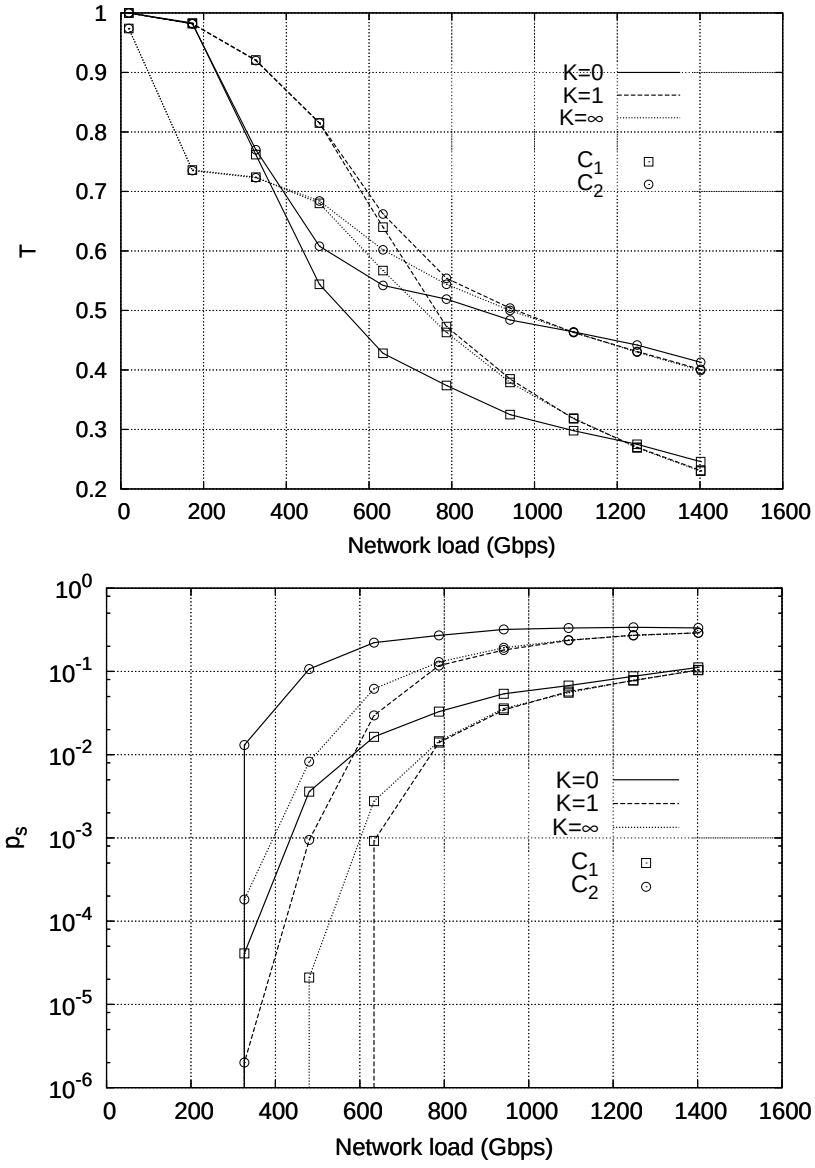


Figure 7.16: Evaluating bandwidth requests fairness for *R8* with $W = 8$. Average normalized throughput T (upper plot) and starvation probability p_s (lower plot) users requesting different minimum bandwidth

accepted flows, and it is not possible to distinguish between traffic flows with different minimum bandwidth requirements. In other words, a flow with higher minimum requested rate gets starved with higher probability.

One of the major drawbacks of dynamic network management, as it is well known from Internet experience, is the unfair behavior of the network toward longer connections (both in terms of physical distance and in terms of logical hops). One of the interesting questions is whether the physical topology and available number of wavelengths impacts on the problem. Fig. 7.17 reports results for $W = 4$ (upper plot) and $W = 8$ (lower plot) for *R8* and *MT16*.

On the one hand, it is clear that none of the parameter setting can guarantee perfect fairness, and that the physical topology does not have a major impact. On the other hand, increasing W does help in keeping a certain degree of fairness, specially if the grooming policy tends to build a logical topology that approaches a full mesh ($K = 1$). Instead, the very aggressive use of optical resources operated with $K = 0$ gobbles resources to build unnecessary parallel paths between adjacent node pairs, exacerbating the unfairness at high loads.

An intelligent grooming policy should compensate for, at least in part, this inherent unfairness by *sparing* optical resources to dedicate to longer flows. However, in an overlay model and without costly traffic measurements, this might not be easy to implement. We have defined a *LEngth DEpendent optical closing: LEDE* policy that defines the delay of closure for a lightpath of H physical hops as $(H-1)\tau_{cl}^{b}$, where τ_{cl}^{b} is a common base closing timeout computed as a function of the network load (always greater than 0). As already introduced in Sect. 7.2, this dependency on the load structure can be very useful to avoid excessive oscillations in the logical topology.

The fairness of the three grooming algorithms according to the distance between the end-nodes of a flow, with and without the *LEDE* option as defined above is studied here, by considering *R8* only (with $W = 8$) which is an ideal topology to analyze this behavior since the regularity of the topology does not

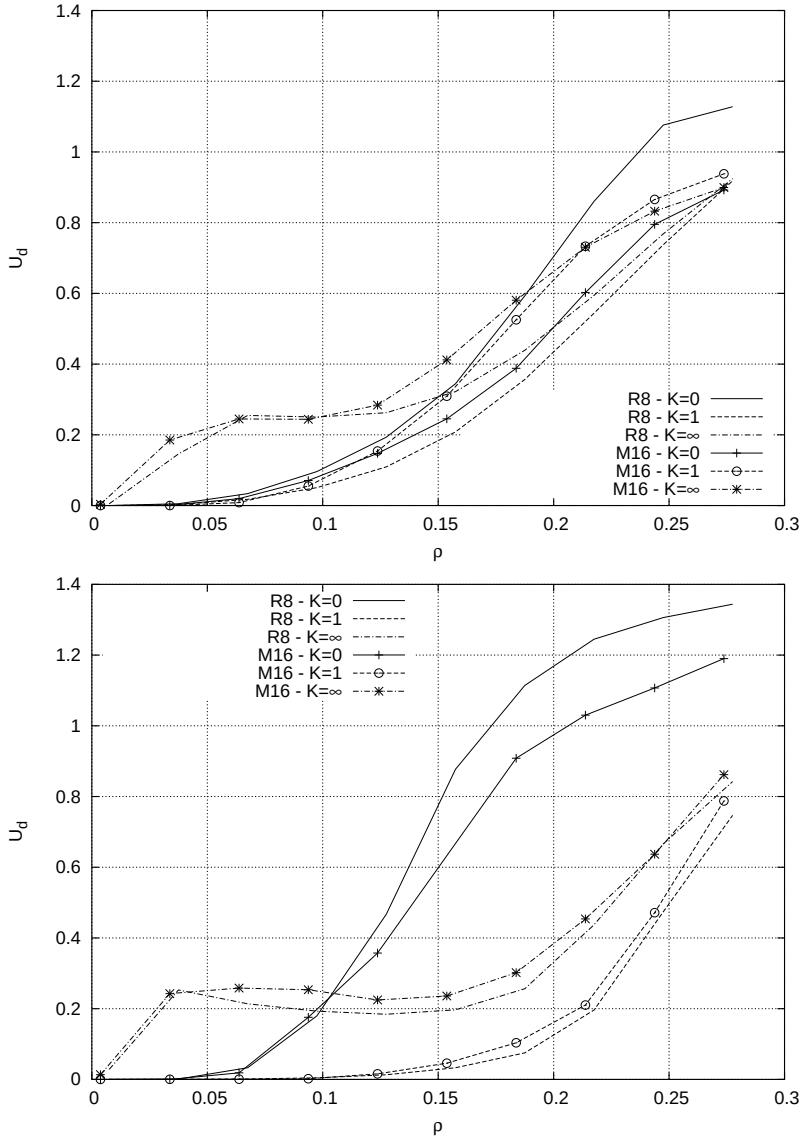


Figure 7.17: Fairness comparison between *R8* and *MT16* with $W = 4$ (upper plot) and $W = 8$ (lower plot)

introduce any distortion effect. When this option is not active, τ_{cl} is set to 0. Otherwise, the base closing timeout τ_{cl}^{b} introduced previously is computed according to the average flow interarrival time[8] and is thus automatically dependent on the network load. In each simulation it is set to $\tau_{cl}^{b} = t_{ia}/2$, where t_{ia} is the mean value of the time spent between the arrival of two flow-requests with the same nodes pair as source and sink.

The upper plot in Fig. 7.18 shows that the *LEDE* option effectively alleviates the unfairness, by uniformly reducing the fairness index. In particular, at very high load (for $p_s \simeq 20\%$), U_d is always below 1 for both $K = \infty$ and $K = 1$. The impact of the *LEDE* option over the average number of links per optical path is well illustrated in the middle plot, which shows an average increase of N_{lo} for all the grooming algorithms. This behavior proves that increasing the closure timeout on longer optical routes increases the amount of resources dedicated to longer routes and reduces the unfairness toward longer flows in IPO networks. The lower plot shows instead a useful 'side-effect' of *LEDE*. In fact, keeping lightpaths not carrying active traffic in the network open for longer periods, the frequency of routing table updates due to virtual topology changes is reduced. Starvation probability and throughput (not shown for the sake of brevity) show that this is obtained without average performance losses.

7.4.7 Single layer TE and comparison with static resource assignment

So far we have limited the discussion to the performance of dynamic grooming policies. However, two major questions remains open: how dynamic grooming compares with optimal, static resource assignment, and how a dynamic grooming policy interacts with constraint-based routing algorithms in either the optical or IP layer.

Constraint-based routing are well known techniques to perform TE in IP networks. Analogously, we can consider "adaptive routing" in wavelength routed

[8] Using management information it can be easily implemented in real networks.

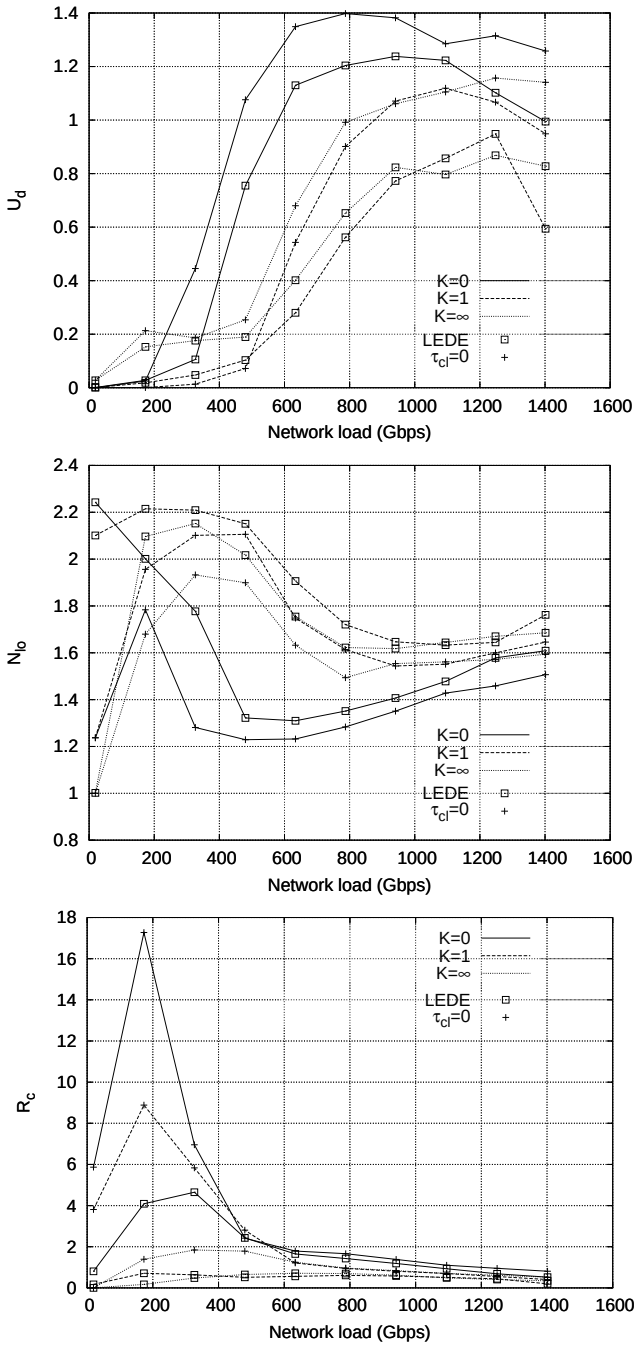


Figure 7.18: Fairness comparison for grooming algorithms with and without *LEDE* option for *R8* with $W = 8$: impact over the distance unfairness index U_d, the average number of links per optical path N_{lo} and the routing table change rate R_c.

networks as TE techniques for the optical layer, since the route from a source to a destination is chosen dynamically depending on the network status.

For the sake of simplicity, and also because of the complexity of optimal static resource assignment in general topologies, we limit the study to the *R8* topology with $W = 8$. For such network it is possible to build up a full mesh in the virtual topology using the method presented in [122]. In case of uniform traffic this is an optimal solution of static resource assignment.

The static grooming acts as reference. We consider here three possible solutions, all based on ***HC*** with $K = 1$ and $\tau_o = 3$ Gbit/s.

A) Both $\mathbf{\Lambda}$ and $\mathbf{\Omega}$ are FSP and in $\mathbf{\Lambda}$ the wavelength assignment is first fit; this combination is named "NO TE" since the routing algorithm is the simplest possible at both levels.

B) $\mathbf{\Lambda}$ is FPLC with first fit and $\mathbf{\Omega}$ is FSP; this is the combination we used in previous results and it is named TE($\mathbf{\Lambda}$).

C) $\mathbf{\Lambda}$ is FSP with first fit and $\mathbf{\Omega}$ is the MD algorithm described in Sect. 7.1.1; this is named TE($\mathbf{\Omega}$)[9].

Fig. 7.19 reports T and p_s for the four grooming described above. In order to enhance the differences and make comparisons easier, T_{rel} in the upper plot is normalized with respect to the static grooming case, so that this solution throughput is constant and equal to 1. As expected the static, optimal assignment performs generally better than the others, although TE($\mathbf{\Omega}$) achieves higher throughput with lower p_s at light to medium loads, but shows a sharp degradation as the load increases, a behavior often observed with congestion based routing algorithms at the IP level.

In order to complete the picture we report results for a case of time varying traffic, where we expect that the ***HC*** grooming will dynamically adapt to the traffic change and improve performance above what is achievable with a static

[9]We deem that the use of TE techniques at both layers without exchanging informations, thus adopting an augmented or peer model, will lead to "conflicts" in the use of resources creating instabilities in the network; however, this issue requires additional investigations.

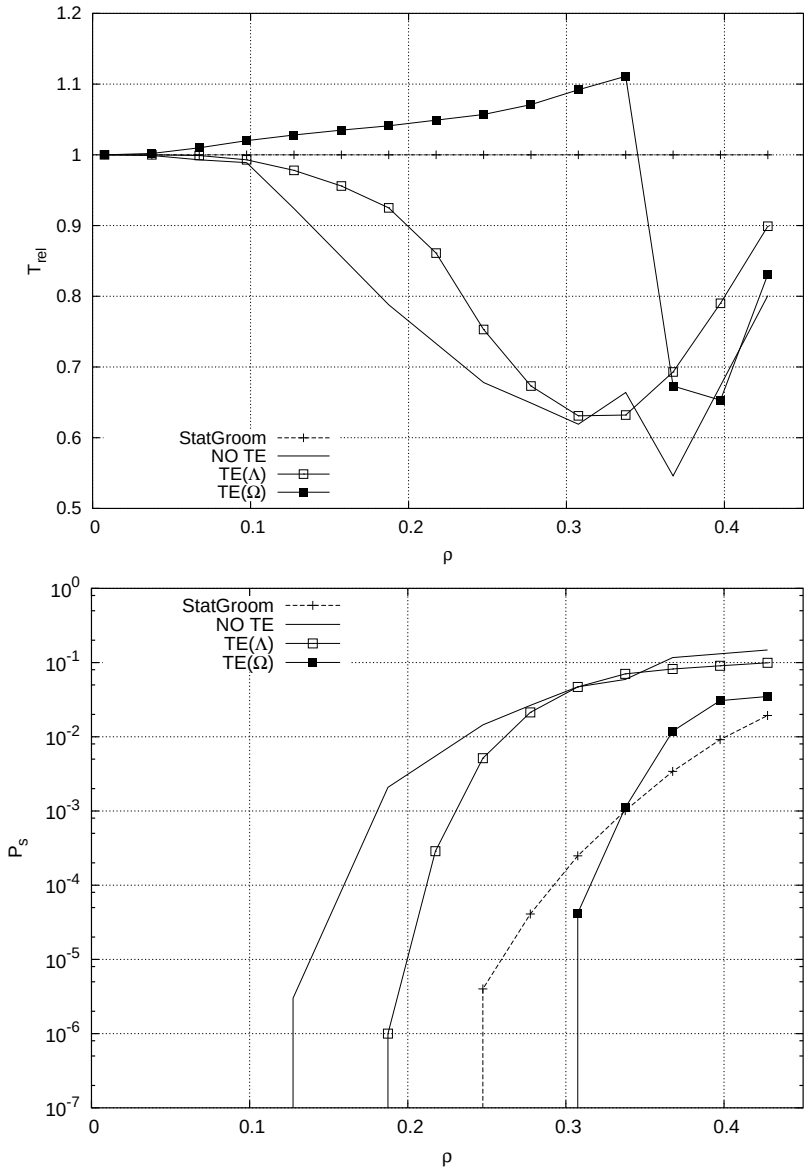


Figure 7.19: Normalized per-flow average throughput T (upper plot) and starvation probability p_s (lower plot) for *R8* with W=8

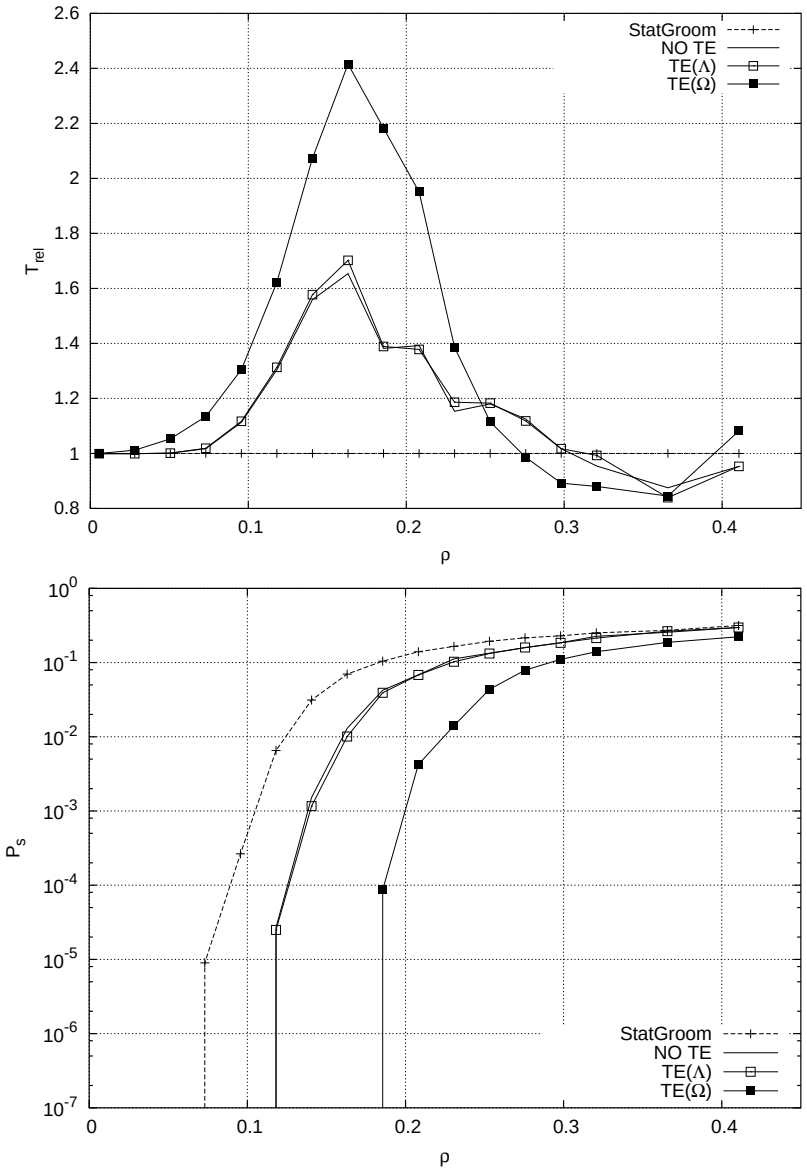


Figure 7.20: Normalized per-flow average throughput T of servers (upper plot) and starvation probability p_s (lower plot) for *R8* with W=8: time varying traffic

solution unable to adapt the resource assignment. On the scenario just analyzed we modified two nodes with distance 4 on *R8*, so that they periodically increase the volume of traffic they generate mimicking the behavior of some server that "switches" on and off the network. The traffic increase is five times and the on and off periods are independent between the two servers, and exponentially distributed with average one hour. Fig. 7.20 reports T_{rel} and p_s for the two servers in this scenario. As expected the relative merit of static and dynamic grooming are reversed, with TE($\mathbf{\Omega}$) remaining the best choice among dynamic grooming. It must be noted that even at high loads, where T_{rel} decreases below one, p_s remains lower than the reference static grooming.

7.5 Conclusions

One of the main contribution of this chapter is the definition of a formalism to describe dynamic grooming policies, defining the limits between grooming in overlay architectures and grooming in peer or augmented architectures, where there is total or partial integration of the optical and IP control planes. A family of grooming policies ***HC*** for the overlay model, based on constraints on the number of hops and bandwidth available at the virtual topology level, is defined as well. Several grooming policies previously presented in literature are particular cases of the family here defined.

Simulation results, supported by heuristic considerations and a very simple analytical model highlighting the interaction effects, show that ignoring the two layer interaction is not correct. In particular, the impact of realistic *elastic* traffic on dynamic grooming is dramatic, showing clearly that approximating IP traffic with CBR-like traffic can lead to wrong conclusions when routing and grooming are considered. The different performance induced in the network by the elastic traffic is such that conclusions drawn with traditional traffic models can be completely misleading.

Results analysis proves that it is possible to define grooming parameters that lead to good performance regardless of the topology and that allow good scaling with the amount of optical resources. The inspection of the virtual topology that are build by the grooming policies hints to the fact that a good policy should try to build a full mesh at the virtual topology level to keep the balance between the traffic pattern (uniform) and the virtual topology.

Moreover, it is shown that the presence of a double network layer (optical and IP) does not alleviate traditional fairness problems associated with best-effort, elastic traffic. Some form of compensations are possible through the use of smart grooming policies; however, in an overlay model, where no information is shared between the optical and the IP level, it is not easy to find the appropriate and definitive solution.

The impact of traffic engineering techniques applied at the optical or IP level was discussed, highlighting that constrained based routing techniques in the IP level, which is characterized by quicker dynamics, ensures better performance with respect to the use of adaptive routing in the optical level. In any case the well known problem of performance degradation at high loads when constrained-based routing is used is present also in IPO networks.

Chapter 8

Conclusions

Within this dissertation, new Traffic Engineering (TE) techniques for the emerging dynamic optical networks are proposed and analyzed. TE is a fundamental solution to optimize the performance of a network and to control the network congestion through an efficient utilization of network resources. In wavelength-routed optical networks, the traffic coming from upper layers such as IP, ATM, MPLS or SONET/SDH is carried over the logical topology defined by the set of established lightpaths. The main objective of TE for optical networks is the optimization of optical resources configuration with respect to a particular traffic demand, which can be obtained through two main techniques: Virtual Topology Reconfiguration (VTR) and Traffic Grooming (TG).

In chapter 4 new load balancing techniques for the VTR problem are proposed based on IP-like routing, where the forwarding mechanism is only driven by the destination address. The proposed algorithm (RSNE — Reverse Subtree Neighborhood Exploration) implements a Local Search technique where the basic step is the modification of a single entry in the routing table of a node. A randomized method with reduced computational complexity (fRSNE — fast RSNE) is also presented and analyzed. Both schemes allow an incremental implementation where local search steps are continuously performed as traffic conditions change. Experiments under static and dynamic (time varying)

traffic scenarios show a rapid reduction of the congestion. The performance of the incremental scheme while tracking a changing traffic matrix is comparable to that obtainable through the complete re-optimization of the traffic, while the randomized implementation is particularly efficient when scaling properties are considered.

Many extensions to the proposed schemes can be envisioned, in particular when considering specific properties of the optical medium. At the moment these schemes could work properly on Optical Packet Switching networks where IP-like routing is assumed or in wavelength routed networks where wavelength conversion at each node is considered. The context of networks having links with different capacities should also be considered in future extensions, as well as more general routing mechanisms (e.g. G-MPLS). Further investigation will determine how the randomized version could be exploited in a distributed environment, where complete information is not available at each node, in order to have a fast implementation on an asynchronous network.

Two novel Traffic Engineering (TE) schemes for congestion control in MPLS networks are discussed in chapter 5. While most existing TE schemes to prevent network congestion rely on constraint-based routing (CBR), the proposed algorithms use a Local Search technique where the basic move is the modification of the route for a single Label Switched Path (LSP). Because modifications cause a temporary disruption in the network, a “laziness” criterion implies that moves are executed only when absolutely necessary or when the situation is very close to requiring it. Experiments under a dynamic traffic scenario show a reduced rejection probability especially with long-lived and bandwidth consuming connection requests, thus proving a better network resource utilization compared to existing CBR schemes in MPLS networks, while guaranteeing a reduced computational complexity.

In chapter 6 a new TE scheme to efficiently route sub-wavelength requests with different QoS requirements (TG problem) is proposed for IP over Optical

networks. In most previous studies on TE based on dynamic traffic grooming, the objectives are to minimize the rejection probability by respecting the constraints of the optical node architecture, but without considering service differentiation. In practice, some high-priority (HP) connections can instead be characterized by specific QoS constraints on the maximum tolerable end-to-end delay and packet-loss ratio. In the proposed solution, when a new request arrives, an on-line grooming scheme finds a route which fulfills the QoS requirements. If a HP request is blocked at the ingress router, a preemption algorithm is executed locally in order to create room for this traffic. The proposed preemption mechanism minimizes the network disruption, measured by the number of rerouted low-priority connections and new set-up lightpaths, and reduces the signaling complexity.

Future improvements of the proposed scheme should consider a more accurate representation of the network which considers the number of ports per G-OXC in the network or even the specific physical properties of the optical devices (fibers, OXCs, amplifiers,...) in order to better evaluate the real impact of specific lightpaths to guarantee the transmission quality of high-priority connections. Furthermore, by considering the two highlighted constraints, delay and packet-loss, it is possible to consider more intermediate classes of traffic: it could be interesting to study the impact of different preemption policy when more than two class types are considered in the network.

Finally, a formal framework for the definition of dynamic grooming policies in IP over Optical networks is defined in chapter 7. The formal framework is then specialized for the Overlay Architecture, where the control plane of the IP and optical level are separated, and no information is shared between the two. For the first time to the best of our knowledge, the performance analysis of a parametric family of grooming policies for the Overlay Architecture is evaluated by considering a Dynamic Statistical Multiplexing (DSM) grooming approach. Results are derived by using realistic traffic models that depart from

the circuit-like traffic traditionally used in grooming studies, and fairness issues versus the flow physical length are also discussed.

Results analysis proves that it is possible to define grooming parameters that lead to good performance regardless of the topology and that allow good scaling with the amount of optical resources. The inspection of the virtual topology that are build by the grooming policies hints to the fact that a good policy should try to build a full mesh at the virtual topology level to keep the balance between the traffic pattern (uniform) and the virtual topology. This observation may pave the road for the definition of more performing strategies that the ones we analyzed, which can also adapt to asymmetric and time varying traffic.

Appendix A

GANCLES: A Tool to Study Dynamic Grooming in IPO Networks

Dynamic grooming of IP traffic over a wavelength routed optical network means that the two routing layers (IP and optical) interact, with deep impacts on Traffic Engineering (TE) and QoS provisioning. The interaction nature depends on the grooming algorithm, as well as on the amount of information (if any) exchanged between the two layers. This situation is very complex and its study is normally done via simulations with a modelling effort to reduce the problem complexity, e.g., without simulating packet level traffic, but with fluid models. Sivalingam et al. have presented an *ns*-2 based simulation tool for performance studies of WDM networks [115]. This simulator does not consider the problem of grooming and the WDM management layer is seen as a logical layer on top of an IP network (the standard *ns*-2 network layer) building virtual circuits on the packed switched routing layer.

In this appendix a novel tool for the study of IP over Optical networks is presented. The tool, freely available on-line [5], is a network level simulator named GANCLES that includes several innovative features allowing the study of realistic scenarios in IP over Optical networking, making it an ideal tool for Traffic Engineering purposes.

A.1 The Tool Features and Architecture

GANCLES is an event-driven asynchronous simulator derived from ANCLES [1]. ANCLES has gradually evolved over the years to allow the simulation of elastic connections over IP networks, with a flow-level granularity, as described in [24]. A separate extension allows the study of different lightpath-level granularity in ASON-based wavelength-routed networks [2].

GANCLES integrates the two network layers (from now on: the *data-layer* and the *optical-layer*), thus allowing the in-depth study of the interaction between them when a multi-layer network environment such as IP over WDM is considered. The objective of the simulator is to give researchers a useful tool to study new algorithms and protocols, to analyze network performance, to implement traffic engineering criteria, and to design QoS provisioning means in this multi-layer environment.

The simulator includes advanced tools to perform statistical analysis based on the "batch-means" technique [85]. Simulation experiments are stopped when the desired user-specified accuracy is reached on a selected set of performance parameters.

The tool allows the computation of a large number of performance indexes, both for the entire network and for selected traffic relations.

The network models simulated by the tool are composed of instances of three basic entities.

- NODES, which perform the routing functions at the IP and at the WDM layer, and implement the CAC and the grooming algorithms; NODES can be either Non-Grooming OXCs, switching entire lightpaths, or Multi-hop full Grooming OXCs, where the electronic fabric is based on a IP router allowing for data-layer traffic injections/extraction and performing grooming operations[1]; these nodes are named G-OXC for simplicity.

[1]Other NODES architectures like Multi-hop partial Grooming OXC (see Sect. 3.3) are currently under imple-

Two node architectures are considered here (see Sect. 3.3): a node can be a Non-Grooming OXC, which allows to switch entire lightpaths from an ingress port to an egress port, or it can be a Multi-hop full Grooming OXC (G-OXC), where the electronic fabric is based on a IP/MPLS router. These nodes can both terminate transit traffic or they can groom it into some optical pipe with incoming IP traffic. In the rest of this appendix the terms (IP) router, LSR (Label Switching Router), and G-OXC are used interchangeably.

- CHANNELS, that accommodate the information transfer between either adjacent NODES or USER-NODE pairs; a CHANNEL can accommodate up to W independent lightpaths.

- USERS, that acts as sources and sinks for the traffic flowing through the network; USERS can be both at the optical-layer, generating "circuit-like" requests of entire lightpaths, and connected directly to OXCs, or at the data-layer, connected to G-OXCs only and generating sub-wavelength requests that can follow one of three different "models:" i) traditional circuit-like requests, ii) Time Based (TB) best-effort requests, and iii) Data Based (DB) best-effort requests (See Sect. A.3 for further details).

Simulations are specified with a formal grammar inherited from [1] and named ND. The number of CHANNELS and their data rates are expressed in number of fiber per link, number of wavelengths per fiber and finally, transmission capacity in Gbit/s per wavelength. The NODE architecture is described in detail discussing the interaction between the data-layer and optical-layer in Sect. A.2.

Due to its specific implementation characteristics, GANCLES allows to have full, partial or no exchange of information between the data-layer and the optical-layer, therefore allowing the description of any of the grooming algorithms (overlay or integrated, see Sect. 2.3.1) proposed so far in literature.

mentation.

A.2 Physical and Logical Topology Management and Interaction

As mentioned before, in GANCLES nodes can be pure OXCs, which allows switching entire lightpaths from an ingress port to an egress port, or they can be G-OXCs, which supports sub-wavelength traffic flows and multiplex them onto wavelength channels through a grooming fabric. OCXs and G-OXCs can be mixed freely into a simulation experiment. It is possible to have both *full opaque* or *full transparent* crossconnects. Opaque OXCs allows full wavelength conversion; transparent OXCs have no conversion capabilities. Partially opaque OXCs, with limited conversion capabilities, are being implemented. A G-OXC is *also* a router, hence the transit traffic (not terminated in the router), can be groomed with incoming traffic. Sub-wavelength traffic can be generated and received only in G-OXCs.

When considering a multi-layer environment, with connections flowing from IP level users through an optical network, we need to distinguish between a *physical* topology and *logical* topology. The latter one is made of all the lightpaths established between G-OXCs over the physical topology according to some optical-level routing algorithms. The logical topology is used for routing at the data-layer (IP) and it is modified each time the grooming management entity triggers the establishment or release of some lightpaths.

Simulation in GANCLES are driven by the USERS, which collect requests from their associated call generators and forward them to the network, while acting as destinations for the connections coming from remote users.

Fig. A.1 represents the interaction between different GANCLES parts. A simulation starts after GANCLES has acquired the simulation experiment description in ND. The description includes: (i) the network topology in terms of a weighted graph connecting USERS and NODES through CHANNELS; (ii) the traffic relations between USERS and the statistical characterization of sources;

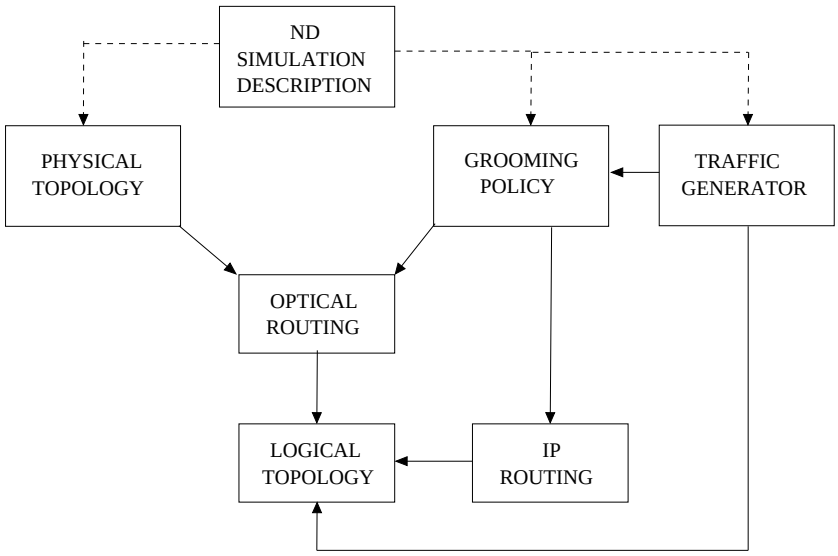


Figure A.1: Logical interaction between different high-level modules in GANCLES, the management of the optical-layer is mediated by the grooming strategies and algorithms

(iii) the selection of the routing, CAC and grooming algorithms adopted for the simulation run; (iv) a number of options concerning both the network operation and the simulation session management; (v) the performance indices to be measured.

When the simulation starts an optical-layer function independent from the selected RWA algorithm defines the set of available physical paths for each pair of OXCs/G-OXCs ordering them according to some specific criterion. Only this set can be used by the routing algorithm, reducing the routing problem complexity.

Every time a new lightpath is added or removed from the network by the optical-layer, the data-layer (logical) topology is changed. As done at the optical-layer, also in this case the set of possible logical paths between each G-OXC (router) pair is computed following a user-specified criterion. This task is extremely critical, because path-computing is very time consuming, hence the path selection criterion must be carefully defined depending on the data-layer routing algorithm selected (e.g., only one path needs to be computed if Fixed

Shortest Path routing is used).

Notice that the overall setup (and performance) of IPO networks is heavily influenced by technology constraints. We already mentioned opaque or transparent OXCs, but other constraints, such as simplex- or duplex-only lightpath management also come into play at the optical-layer. Similar constraints may arise if TE techniques are used at the data-layer. GANCLES allows the simulation of an arbitrary mix of these constraints.

Each time a USER generates a connection request, the grooming entity decides whether: (i) route it over the current topology; or (ii) ask the optical-layer to open one or more lightpaths (thus modifying the logical topology) and then route the request at the data-layer. In the first case, a data-layer CAC algorithm (if any) is executed for each of the paths being considered by the data-layer routing algorithm; if no path is found to route the incoming request in agreement with its QoS requirements, the request is dropped. In the second case, after the logical topology modification, the same data-layer routing and CAC are applied. As a general rule, in this case a connection can be refused only if the optical-layer was not able to modify the logical topology meeting the grooming algorithm requirements. For each request the grooming procedure may require an arbitrary mix of actions of type (i) and (ii), depending on the complexity of the algorithm implemented and on the model (peer, augmented, overlay) assumed for the IPO.

The interaction between the optical-layer and the data-layer lies at the core of dynamic grooming problems. This interaction is described in GANCLES (as in real networks) by the optical/IP control interface within G-OXCs. This simple and clear, though realistic, interface implementation is one of the innovative features of GANCLES.

Modifying the logical topology, the question arises whether all the traffic is re-routed or if only new connections follow the new available routes. GANCLES presently assumes the second option, but its modification for re-routing

is trivial.

When an active connection terminates, the corresponding resources in the logical topology are released. This operation can lead to the release of some lightpaths if they are not carrying traffic anymore. In this case, paths at the IP level need to be recalculated again over the new logical topology. The specific lightpath releasing criteria (e.g., a given time-lapse without traffic) can be specified by GANCLES user.

GANCLES implicitly assumes the utilization of a separated control plane (e.g., GMPLS) to keep each node informed of the network status. The control traffic is not considered in the performance measures.

A.3 Traffic Sources

The present release of GANCLES provides different types of call generators at both the data-layer and optical-layer. Each connection request is associated with the identifier of the destination USER, which is chosen accordingly to the traffic relations specified for the experiment.

The main traffic models implemented for the data-layer USERS are: CBR, ON-OFF CBR, Uniform VBR, Video VBR or Best-Effort with TB and DB model (see [1]), while for the optical-layer USERS it is possible to generate lightpath requests according to Poisson traffic generators, but it is also possible to generate permanent and semi-permanent lightpaths (see [2]), a feature that enables adding constraints or static portions to the logical topology.

Since GANCLES has been developed mainly to study the interaction between IP (as data-layer) and the optical-layer, a more detailed explanation of the *elastic* traffic features of Best-Effort USERS is needed. As a general rule IP does not implement CAC functions and the congestion control is done reactively by TCP. As we discussed in chapter 7 the elastic nature of present-day data applications cannot be disregarded if dynamic grooming is considered, because

the feedback introduced by the closed-loop nature of TCP (and traffic aggregation does not destroy the feedback) has an enormous impact on the overall performance. We introduce two different models of elastic traffic. Both share the characteristic that a flow f_{sd} arrives to the network with a backlog of data D to transmit and both include some form of elasticity, though very different one another.

The first model, named *time-based* (TB), assumes that the elasticity is taken into account only reducing the transfer rate when congestion arises. The flow duration is determined when the flow arrives to the network, based on its backlog D and its peak transmission rate B_M (e.g., the access link speed) $\tau = \frac{D}{B_M}$. The effect of congestion is just that the throughput of flows is reduced, but their closing time is not affected. A consequence of this behavior is that the data actually transferred by a flow f_{sd} is generally less than the "requested" amount D, thus reducing the actual network load and relieving congestion. This model is very simple and does not grab all the complexity of the closed-loop interaction between the sources and the network.

A more accurate model, named *data-based* (DB), assumes instead an ideal max-min sharing of the resources within the network at any given instant. Flows still arrive to the network with a backlog D and with a peak transmission rate B_M. The acceptance of a new flow will affect not only all the other flows on the same path, but indeed all the flows in the network, since the max-min fair share is completely recomputed updating the estimated closing time of all the flows in the network. The same applies when flows close, freeing network resources. This model includes the most important feature of elastic traffic, which is the positive feedback on the flows duration. The more congested is the network, the longer the accepted flows remain in the network.

Without a CAC, at high loads the network can become instable, as the number of flows within the network can grow to infinity and their individual throughput goes to zero. To build a more realistic scenario, a second attribute has been

introduced to characterize any IP flow f_{sd}: a minimum requested rate b_m. If at any time the bandwidth assigned to f_{sd} falls below it, then flow closes and is counted as a "starved" flow, because the network was not able to guarantee its correct completion. The attributes b_m and B_M are included in some SLA (Service Level Agreement) at the IP/Optical interface (see [46] for initial works on Optical-SLA).

A new performance measure can therefore be introduced, called the *starvation probability* p_s, which complements more traditional metrics such as throughput, blocking probability, optical-layer overhead in opening and closing lightpaths, etc.

A.4 Routing Algorithms

Inheriting the terminology in [1], whatever criterion is used to order the paths at both the optical-layer or data-layer, a *primary path* is always defined for each pair of NODES. All the other allowed paths are referred to as *secondary* paths.

Data-layer Routing Several alternatives are available to route calls at data-level so as to take into account the dynamic load of the network. An important property of GANCLES is the possibility to simulate both source-based (e.g., MPLS-like) or hop-by-hop routing.

The following list enumerates the main routing algorithms implemented in GANCLES. The reader is referred to [1] for more details on this part and the relative references to the literature.

- *Single Path Routing*: only the primary path between the nodes is considered to route the connection. This is also known as Fixed Shortest Path.
- *Controlled & Uncontrolled Alternate Routing*: if there is no space for the connection along the primary path, the secondary ones are investigated with different constraints (see [103] for details).

- *Minimum Distance Routing*: for each source-destination pair, the path π is chosen that minimizes the following quantity: $C_\pi = \sum_{l \in \pi} \frac{1}{b_l}$ where b_l is the max-min fair bandwidth that is available to a new connection over link l belonging to path π (see [73] for details).

- *Widest-Shortest Routing*: this algorithm first identifies the minimum-hop-count paths and breaks ties by choosing, among the paths with the minimum hop count, the one with the maximum available bandwidth (see [113] for details).

Optical-layer Routing There are different static and dynamic algorithms implemented for the routing of optical-layer requests. In the following only single-path routing algorithms are described, but the tool provides also for protective routing, with both *dedicated* or *shared* protection mechanisms (see [2] for more details).

If we use opaque OXCs, the wavelength continuity is necessary only in the transparent sections, i.e. the part of route connecting two opaque nodes. Allowing wavelength conversion in the opaque nodes implies that the wavelength assigned to the connection can be different in each transparent section. The main optical routing algorithms implemented in GANCLES are:

- *Fixed Shortest Path*: it routes the lightpath request always on the dedicated primary path between the endpoints.

- *Shortest-Widest Path*: it selects the paths with the largest number of available optical channels; if there are more possibilities, it routes the lightpath on the shortest among them.

- *Alternate Shortest Path*: it selects the shortest from those paths where there is at least one wavelength available.

Furthermore, Wavelength Assignment (WA) algorithms can be freely associated to any routing. WA algorithms include *Random* and *First-Fit*, and others

can be added easily if required (details and references on optical-layer routing and RWA can be found in [2]).

A.5 Grooming Algorithms

Grooming policies are the ensemble of algorithms and protocols that take the decisions regarding possible changes of the current logical topology each time a data-layer request arrives or leaves the network. When new logical links need to be installed, two factors must be determined: how many of them must be set up and between which nodes.

The decision to route the incoming requests over the existing logical topology or to establish new lightpaths to create more room for them can lead to different network performances. A general analysis of different "grooming policies" is carried out in [126] under the hypothesis of bandwidth-guaranteed traffic within a peer IPO model. When *elastic traffic* is considered, there is no obvious upper limit to the possible number of flows which is routed onto the existing logical layer. In this case, the need for the establishment of new lightpaths must be introduced based on some suitable parameter τ_o called *optical opening threshold* as described in Sect. 7.2.

In the current version of the simulator only the parametric family of overlay grooming policies ***HC*** described in chapter 7 is implemented, therefore the main parameters to introduce in GANCLES to run a simulation on dynamic grooming are: $\mathbf{\Lambda}$, $\mathbf{\Omega}$, K and τ_o.

Note that in all these cases if the source-destination pair is disconnected on the logical topology and no new lightpath can be installed between them, the incoming connection request is refused even if it is elastic. This phenomenon is particularly evident at low/medium loads with grooming algorithms using aggressively the optical resources, as it was shown in Sect. 7.4.4. Such unacceptable situations could be avoided by including a pre-defined logical topology

$\mathcal{P}_{pre}$ of pre-established lightpaths to ensure the connectivity in the data-layer, a feature GANCLES is provided with, defining different basic logical topologies (such as spanning tree, full mesh,...).

When a lightpath needs to be released because it is not carrying traffic, the simulator gives the possibility of delaying the closure using a timeout, called *optical closing time-out* τ_{cl}. When no traffic is carried over some lightpath, it is kept open for the timeout period and gets closed only if its state does not change.

Appendix B

Publications authored

Publications produced during the period that work for this PhD thesis was carried out:

B.1 Peer-reviewed journal papers

M. Brunato, R. Battiti and **E. Salvadori**. Dynamic Load Balancing in WDM Networks. *Optical Networks Magazine*, Special issue on: *Dynamic Optical Networking: Around the Corner or Light Years Away?*, 4(5):7–20, 2003.

B.2 Conference papers

M. Brunato, R. Battiti and **E. Salvadori**. Load Balancing in WDM Networks through Adaptive Routing Table Changes. In *Networking*, number 2345 in Lecture Notes in Computer Science, pages 289–301, Pisa - Italy, May 2002. Springer Verlag.

E. Salvadori and R. Battiti. A Load Balancing Scheme for Congestion Control in MPLS Networks. In *IEEE Symposium on Computers and Communications - ISCC*, pages 951 – 956, Antalya, Turkey, July 2003.

E. Salvadori and R. Battiti. A Traffic Engineering Scheme for QoS Routing in G-MPLS Networks Based on Transmission Quality. In *IFIP Proceedings of*

the 8th ONDM, Gent, Belgium, February 2004.

E. Salvadori and R. Battiti. Quality of Service in IP over WDM: Considering Both Service Differentiation and Transmission Quality. In *IEEE Proceedings of ICC*, pages 1836 – 1840, Paris, France, 2004.

R. Lo Cigno, **E. Salvadori** and Z. Zsóka. Elastic Traffic Effects on WDM Dynamic Grooming Algorithms. In *IEEE Proceedings of GLOBECOM*, Dallas, Texas, USA, December 2004.

E. Salvadori, Z. Zsóka and R. Lo Cigno. Dynamic Grooming in IP over WDM Networks: A Study with Realistic Traffic based on GANCLES Simulation Package. In *IFIP/IEEE Proceedings of the 9th ONDM*, Milan, Italy, February 2005.

B.3 Submitted papers

E. Salvadori, Z. Zsóka, R. Lo Cigno and R. Battiti. A Framework for Dynamic Grooming in IPO Overlay Architectures. Submitted for publication, 2005.

H. Abrahamsson, S. Balon, S. Bessler, M. D'Arienzo, O. Delcourt, J. Domingo-Pascual, S. C. Erbas, I. Gojmerac, G. Leduc, A. Pescapè, B. Quoitin, S.P. Romano, **E. Salvadori**, F. Skivée, H.T. Tran, S. Uhlig and H. Umit. An Open Source Traffic Engineering Toolbox. *Computer Communications*, 2005. Submitted for publication.

B.4 DIT Technical reports

E. Salvadori, R. Battiti, and F. Ardito. Lazy Rerouting for MPLS Traffic Engineering. Technical report, Università di Trento, Dipartimento di Informatica e Telecomunicazioni, March 2003. DIT-03-011.

E. Salvadori and R. Battiti. Traffic Engineering in G-MPLS networks with

QoS guarantees. Technical report, Università di Trento, Dipartimento di Informatica e Telecomunicazioni, October 2003. DIT-03-050.

E. Salvadori, Z. Zsóka, D. Severina and R. Lo Cigno. GANCLES: A Network Level Simulator to Study Optical Routing, Wavelength Assignment and Grooming Algorithms. Technical report, Università di Trento, Dipartimento di Informatica e Telecomunicazioni, April 2004. DIT-04-017.

R. Lo Cigno, **E. Salvadori** and Z. Zsóka. Elastic Traffic Effects on WDM Dynamic Grooming Algorithms (extended version). Technical report, Università di Trento, Dipartimento di Informatica e Telecomunicazioni, August 2004. DIT-04-091.

Bibliography

[1] ANCLES - A Network Call-Level Simulator. http://www.tlc-networks.polito.it/ancles.

[2] ASONNCLES - ASOn Network Call-Level Simulator. http://www.hit.bme.hu/~zsoka/asoncles.

[3] CA*net4 - CANARIE Inc. http://www.canarie.ca.

[4] Common Control and Measurement Plane (ccamp) working group's web page. http://www.ietf.org/html.charters/ccamp-charter.html.

[5] GANCLES - Grooming cApable Network Call-Level Simulator. http://netmob.unitn.it/tools/gancles.

[6] International Telecommunication Union. http://www.itu.int.

[7] Internet Engineering Task Force. http://www.ietf.org.

[8] Optical Internetworking Forum. http://www.oiforum.com.

[9] P. Ashwood-Smith and L. Berger. GMPLS Signaling Constraint-based Routed Label Distribution Protocol (CR-LDP) Extensions. IETF RFC 3472, January 2003.

[10] C. Assi, A. Shami, M. A. Ali, Y. Ye, and S.Dixit. Integrated Routing Algorithms for Provisioning "Sub-Wavelength" Connections in IP-

Over-WDM Networks. *Photonic Network Communications*, 4:377–390, July/December 2002.

[11] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. Overview and Principles of Internet Traffic Engineering. RFC 3272, May 2002.

[12] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for Traffic Engineering Over MPLS. IETF RFC 2702, September 1999.

[13] D. Awduche and Y. Rekhter. Multiprotocol lambda switching: combining MPLS traffic engineering control with optical crossconnects. *IEEE Communications Magazine*, 39(3):111–116, March 2001.

[14] A. Banerjee, J. Drake, J.P. Lang, B. Turner, D. Awduche, L. Berger, K. Kompella, and Y. Rekhter. Generalized Multiprotocol Label Switching: an Overview of Signaling Enhancements and Recovery Techniques. *IEEE Communications Magazine*, 39(7):141–151, July 2001.

[15] A. Banerjee, J. Drake, J.P. Lang, B. Turner, K. Kompella, and Y. Rekhter. Generalized Multiprotocol Label Switching: an Overview of Routing and Management Enhancements. *IEEE Communications Magazine*, 39(1):144–150, January 2001.

[16] G. Banerjee and D. Sidhu. Comparative Analysis of Path Computation Techniques for MPLS Traffic Engineering. *Journal of Computer Networks*, 40(2):149–165, September 2002.

[17] L. Berger. GMPLS Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions. IETF RFC 3473, January 2003.

[18] D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall, 1987.

[19] A. Bosco, A. Botta, M. Intermite, P. Iovanna, and S. Salsano. Distributed Implementation of a Pre-Emption Mechanism for a Network Control Based on IP/MPLS Paradigm. In *Proceedings of the 7th IFIP ONDM*, Budapest, Hungary, February 2003.

[20] R. Boutaba, W. Szeto, and Y. Iraqi. DORA: Efficient Routing for MPLS Traffic Engineering. *Journal of Network and Systems Management*, 10(3):309 –325, September 2002.

[21] M. Brunato, R. Battiti, and E. Salvadori. Load Balancing in WDM Networks through Adaptive Routing Table Changes. In *Networking*, number 2345 in Lecture Notes in Computer Science, pages 289–301, Pisa - Italy, May 2002. Springer Verlag.

[22] M. Brunato, R. Battiti, and E. Salvadori. Dynamic Load Balancing in WDM Networks. *Optical Networks Magazine*, 4(5):7–20, 2003.

[23] A. Capone, L. Fratta, and F. Martignon. Virtual Flow Deviation: Dynamic Routing of Bandwidth Guaranteed Connections. In *Proceedings of IEEE Workshop on Quality of Service in Multiservice IP Networks*, pages 608–620, Milan, Italy, February 2003.

[24] C. Casetti, R. Lo Cigno, M. Mellia, M. Munafò, and Z. Zsóka. A Realistic Model to Evaluate Routing Algorithms in the Internet. In *IEEE Proceedings of GLOBECOM*, volume 3, pages 1882–85, San Antonio, Texas, USA, 2001.

[25] C. Casetti, G. Mardente, M. Mellia, M. Munafò, and R. Lo Cigno. Online routing optimization for MPLS-based IP networks. In *Proceedings of HPSR*, pages 215 – 220, Turin, Italy, June 2003.

[26] C. Cavazzoni, V. Barosco, A. D'Alessandro, A. Manzalini, S. Milani, G. Ricucci, R. Morro, R. Geerdsen, U. Hartmer, G. Lehr,

U. Pauluhn, S. Wevering, D. Pendarakis, N. Wauters, R. Gigantino, J.P. Vasseur, K. Shimano, G. Monari, and A. Salvioni. The IP/MPLS over ASON/GMPLS test bed of the IST project LION. *IEEE/OSA Journal of Lightwave Technology*, 21(11):2791–2803, November 2003.

[27] Y. Chen, M. Hamdi, and D.H.K. Tsang. Proportional QoS over WDM networks: blocking probability. In *Sixth IEEE Symposium on Computers and Communications, 2001*, pages 210–215, 2001.

[28] Y. Chen, C. Qiao, and X. Yu. Optical burst switching: a new area in optical networking research. *IEEE Network*, 18(3):16–23, May 2004.

[29] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: A novel approach to high bandwidth optical WANs. *IEEE Transactions on Communications*, 40(7):1171–1182, 1992.

[30] R. Lo Cigno, E. Salvadori, and Z. Zsóka. Elastic Traffic Effects on WDM Dynamic Grooming Algorithms. In *IEEE Proceedings of GLOBECOM*, Dallas, Texas, USA, December 2004.

[31] T. Cinkler. Traffic and λ Grooming. *IEEE Network Magazine*, 17:16–21, March/April 2003.

[32] T. Cinkler and C. Gáspár. Fairness Issues of Routing with Grooming and Shared Protection. In *IFIP Proceedings of ONDM*, pages 665 – 684, Ghent, Belgium, February 2004.

[33] T. Cinkler, D. Marx, C. Popp Larsen, and D. Fogaras. Heuristic Algorithmf for Joint Configuration of the Optical and Electrical Layer in Multi-Hop Wavelength Routing Networks. In *Infocom 2000 Proceedings*, 2000.

[34] M.E. Crovella and A. Bestavros. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, December 1997.

[35] B. S. Davie and Y. Rekhter. *MPLS: Technology and Applications*. Morgan Kaufmann, 2000.

[36] S. De Maesschalck, M. Pickavet, D. Colle, and P. Demeester. Multi-layer Traffic Grooming in networks with an IP/MPLS layer on top of a meshed Optical Layer. In *IEEE Proceedings of GLOBECOM*, pages 2750 – 2754, San Francisco, CA, USA, December 2003.

[37] J.C. de Oliveira, C. Scoglio, I.F. Akyildiz, and G. Ulh. A new preemption policy for DiffServ-aware traffic engineering to minimize rerouting. In *Proceedings of INFOCOM*, pages 695 –704, 2002.

[38] E. Dinan, D.O. Awduche, and B. Jabbari. Analytical Framework for Dynamic Traffic Partitioning in MPLS networks. In *Proceedings of ICC*, volume 3, pages 1604–1608, New Orleans - USA, 2000.

[39] S. Dixit and J. Chrostowski. After the Optical Bubble: The Reality Check. Special Issue of IEEE Communications Magazine, September 2004.

[40] R. Dutta, S. Huang, and G. N. Rouskas. Traffic Grooming in Path, Star, and Tree Networks: Complexity, Bounds, and Algorithms. In *Proceedings of ACM SIGMETRICS 2003*, pages 298–299, San Diego, CA, USA, June 2003.

[41] R. Dutta and G. N. Rouskas. A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks Magazine*, 1(1):73–89, January 2000.

[42] R. Dutta and G.N. Rouskas. Traffic grooming in WDM networks: past and future. *IEEE Network Magazine*, 16:46–56, Nov/Dec 2002.

[43] T.S. El-Bawab and Jong-Dug Shin. Optical packet switching in core networks: between vision and reality. *IEEE Journal on Selected Areas in Communications*, 40(9):60–65, September 2002.

[44] A. Elwalid, C. Jin, S. Low, and I. Widjaja. MATE: MPLS Adaptive Traffic Engineering. In *Proceedings of INFOCOM*, volume 3, pages 1300 –1309, Anchorage - Alaska, 2001.

[45] F. Le Faucheur. Protocol extensions for support of Diff-Serv-aware MPLS Traffic Engineering. IETF Draft, *ddraft-ietf-tewg-diff-te-proto-08.txt*, work in progress, December 2004.

[46] M. Fawaz, B. Daheb, O. Audouin, M. Du-Pond, and G. Pujolle. Service level agreement and provisioning in optical networks. *IEEE Communications Magazine*, 42(1):36–43, January 2004.

[47] Optical Internetworking Forum. User Network Interface (UNI) 1.0 Signaling Specification, December 2001.

[48] L. Fratta, M. Gerla, and L. Kleinrock. The flow deviation method: An approach to store-and-forward communication network design. *Networks*, 3:97–133, 1973.

[49] A.E. Gençata and B. Mukherjee. Virtual-topology adaptation for WDM mesh networks under dynamic traffic. In *Proceedings of 21st Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, 2002.*, volume 1, pages 48–56, 2002.

[50] N. Ghani, S. Dixit, and T.S. Wang. On IP-WDM Integration: A Retrospective. *IEEE Communications Magazine*, 41(9):42–45, September 2003.

[51] F. Giroire, A. Nucci, N. Taft, and C. Diot. Increasing the Robustness of IP Backbones in the Absence of Optical Level Protection. In *Proceedings of INFOCOM*, 2003.

[52] W. Golab and R. Boutaba. Policy-driven automated reconfiguration for performance management in WDM optical networks. *IEEE Communications Magazine*, 42(1):44–51, January 2004.

[53] R. Guerin, D. Williams, and A. Orda. QoS routing mechanisms and OSPF extensions. In *Proceedings of Globecom 1997*, volume 3, pages 1903 –1908, Phoenix, AZ, November 1997.

[54] F. Holness and C. Phillips. Dynamic Congestion Control Mechanism for MPLS Networks. In *SPIE's International Symposium on Voice, Video and Data Communications. Internet, Performance and Control Network Systems*, pages 1001–1005, Boston - MA, November 2000.

[55] I. Iliadis and D. Bauer. A New Class of Online Minimum-Interference Routing Algorithms. In *Networking*, number 2345 in Lecture Notes in Computer Science, pages 959–971, Pisa - Italy, May 2002. Springer Verlag.

[56] P. Iovanna, G. Conte, R. Sabella, M. Settembre, and L. Valentini. A Traffic Engineering Solution for GMPLS Networks: A Hybrid Approach Based on Off-line and On-line Routing Methods. In *Proceedings of the 7th IFIP Working Conference on Optical Network Design & Modelling*, pages 313–332, Budapest, Hungary, February 2003.

[57] A. Jukan and H.R. van As. Service-specific resource allocation in WDM networks with quality constraints. *IEEE Journal on Selected Areas in Communications*, 18:2051–2061, October 2000.

[58] A. Jüttner, B. Szviatovszki, A. Szentesi, D. Orincsay, and J. Harmatos. On-demand Optimization of Label Switched Paths in MPLS Networks. In *Proceedings of IEEE ICCCN*, October 2000.

[59] A. Kaheel, T. Khattab, A. Mohamed, and H. Alnuweiri. Quality-of-Service Mechanisms in IP-over-WDM Networks. *IEEE Communications Magazine*, pages 38–43, December 2002.

[60] K. Kar, M. Kodialam, and T.V. Lakshman. Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications. *IEEE Journal on Selected Areas in Communications*, 18(12):2566 –2579, December 2000.

[61] E. Karasan and E. Ayanoglu. Effects of wavelength routing and selection algorithms on wavelength conversion gain in wdm optical networks. *IEEE/ACM Transactions on Networking*, 6(2):186–196, April 1998.

[62] A.A. Kherani and A. Kumar. Stochastic Models for Throughput Analisys of Randomly Arriving Elastic Flows in the Internet. In *IEEE Proceedings of INFOCOM*, volume 2, pages 1014 – 1023, New York, NY, USA, 2002.

[63] M. Kodialam and T.V. Lakshman. Integrated Dynamic IP and Wavelength Routing in IP over WDM Networks. In *Proceedings of INFOCOM*, volume 1, pages 358 –366, Anchorage - Alaska, 2001.

[64] S. Koo, G. Sahin, and S. Subramaniam. Dynamic LSP Provisioning in Overlay, Augmented, and Peer Architectures for IP/MPLS over WDM Networks. In *IEEE Proceedings of INFOCOM*, Hong Kong, China, March 2004.

[65] J. Labourdette and A. Acampora. Logically rearrangeable multihop lightwave networks. *IEEE Transactions on Communications*, 39:1223–1230, August 1991.

[66] J.-F. P. Labourdette and G. W. Hartand A. S. Acampora. Branch-exchange sequences for reconfiguration of lightwave networks,. *IEEE/ACM Transactions on Communication*, 42(10):2822 – 2832, October 1994.

[67] E. Leonardi, M. Mellia, and M. Ajmone Marsan. Algorithms for the topology design in WDM all-optical networks. *Optical Networks Magazine*, 1(1):35–46, January 2000.

[68] L. Li and A. K. Somani. Dynamic wavelength routing using congestion and neighborhood information. *IEEE/ACM Transactions on Networking*, 7(5):779–786, 1999.

[69] J. Lim and K. Chae. Differentiated Link Based QoS Routing Algorithms for Multimedia Traffic in MPLS Networks. In *Proceedings of IEEE International Conference on Information Networking*, pages 587 –592, Beppu City, Oita - Japan, February 2001.

[70] K.H. Liu. *IP Over WDM*. John Wiley, 2002.

[71] K. Long, Z. Zhang, and S. Cheng. Load Balancing Algorithms in MPLS Traffic Engineering. In *Proceedings of IEEE Workshop on High Performance Switching and Routing*, pages 175 –179, Dallas - Texas, May 2001.

[72] Q. Ma, P. Steenkiste, and H. Zhang. Routing high-bandwidth traffic in max-min fair share networks. In *ACM Proceedings of SIGCOMM 1996*, pages 206–217, Palo Alto, CA, USA, August 1996.

[73] Q. Ma, P. Steenkiste, and H. Zhang. Routing High-Bandwidth Traffic in Max-Min Fair Share Networks. In *Proceedings of ACM SIGCOMM'96*, Stanford, CA, USA, August 1996.

[74] X. Masip-Bruin, S. Sánchez-López, J. Solé-Pareta, J. Domingo-Pascual, and D. Colle. Routing and Wavelength Assignment under Inaccurate Routing Information in Networks with Sparse and Limited Wavelength Conversion. In *IEEE Proceedings of GLOBECOM*, San Francisco, CA, USA, December 2003.

[75] D. Mitra, R. Gibbens, and B. Huang. State-dependent routing on symmetric loss networks with trunk reservations. *IEEE Transactions on Communications*, 41(2):400–411, 1993.

[76] E. Modiano and P. J. Lin. Traffic Grooming in WDM Networks. *IEEE Communications Magazine*, 39(7):124–129, July 2001.

[77] M. Moh, B. Wei, and J.H. Zhu. Supporting Differentiated Services with Per-Class Traffic Engineering in MPLS. In *Proceedings of IEEE International Conference on Computer Communications and Networks*, pages 354–360, Scottsdale - Arizona, October 2001.

[78] A. Mokhtar and M. Azizoğlu. Adaptive wavelength routing in all-optical networks. *IEEE/ACM Transactions on Networking*, 6(2):197–206, April 1998.

[79] P. Molinero-Fernandez, N. McKeown, and H. Zhang. Is IP going to take over the world (of communications)? *ACM SIGCOMM Computer Communications Review*, 33(1):113–119, January 2003.

[80] J. Moy. *OSPF: Anatomy of an Internet Routing Protocol*. Addison-Wesley, 1998.

[81] A. Narula-Tam and E. Modiano. Dynamic load balancing in WDM packet networks with and without wavelength constraints. *IEEE Journal of Selected Areas in Communications*, 18(10):1972–1979, Oct 2000.

[82] X. Niu, W.D. Zhong, G. Shen, and T. H. Cheng. Connection Establishment of Label Switched Paths in IP/MPLS over Optical Networks. *Photonic Network Communications*, 6:33–41, July 2003.

[83] A.E. Ozdaglar and D.P. Bertsekas. Routing and wavelength assignment in optical networks. *IEEE/ACM Transactions on Networking*, 11(2):259–272, April 2003.

[84] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137 –150, April 1994.

[85] K. Pawlikowski. Steady-state simulation of queueing processes: a survey of problems and solutions. *ACM Computer Surveys*, 22(2):123–170, June 1990.

[86] R. Perlman. *Interconnections: Bridges, Routers, Switches and Internetworking Protocols (Second Edition)*. Addison-Wesley, 1999.

[87] M. Peyravian and A.D. Kshemkalyani. Connection preemption: issues, algorithms, and a simulation study. In *Proceedings of INFOCOM*, volume 1, pages 143 –151, 1997.

[88] B. Puype, Q. Yan, D. Colle, S. De Maesschalck, I. Lievens, M. Pickavet, and P. Demeester. Multi-Layer Traffic Engineering in Data-Centric Optical Networks. In *Proceedings of the 7th IFIP Working Conference on Optical Network Design & Modelling*, pages 211–226, Budapest, Hungary, February 2003.

[89] Y. Qin, K. Sivalingam, and B. Li. Architecture and analysis for providing virtual private networks (VPNs) with QoS over optical WDM networks. *Optical Network Magazine*, 2:57–65, March/April 2001.

[90] B. Rajagopalan, Luciani J. and D. Awduche. IP over Optical Networks: A Framework. IETF RFC 3717, March 2004.

[91] B. Ramamurthy, D. Datta, H. Feng, J. Heritage, and B. Mukherjee. Impact of transmission impairments on the teletraffic performance of wavelength-routed optical network . *IEEE/OSA Journal of Lightwave Technology*, 10(17):1713–1723, October 1999.

[92] B. Ramamurthy and B. Mukherjee. Wavelength conversion in WDM networking. *IEEE Journal Selected Areas in Communications*, 16(7):1061–1073, September 1998.

[93] R. Ramaswami and K. N. Sivarajan. Design of logical topologies for wavelength-routed optical networks. *IEEE Journal of Selected Areas in Communications*, 14(5):840–851, 1996.

[94] A. Rodriguez-Moral, P. Bonenfant, S. Baroni, and R. Wu. Optical data networking: protocols, technologies, and architectures for next generation optical transport networks and optical internetworks. *IEEE/OSA Journal of Lightwave Technology*, 18(12):1855–1870, December 2000.

[95] L. H. Sahasrabuddhe and B. Mukherjee. Light-trees: Optical multicasting for improved performance in wavelength-routed networks. *IEEE Communications Magazine*, 137(2):67–73, February 1999.

[96] E. Salvadori and R. Battiti. A Load Balancing Scheme for Congestion Control in MPLS Networks. In *IEEE Symposium on Computers and Communications - ISCC*, pages 951 – 956, Antalya, Turkey, July 2003.

[97] E. Salvadori and R. Battiti. A Traffic Engineering Scheme for QoS Routing in G-MPLS Networks Based on Transmission Quality. In *Proceedings of the 8th IFIP ONDM*, Gent, Belgium, February 2004.

[98] E. Salvadori and R. Battiti. Quality of Service in IP over WDM: Considering Both Service Differentiation and Transmission Quality. In *IEEE Proceedings of ICC*, pages 1836 – 1840, Paris, France, 2004.

[99] E. Salvadori, R. Battiti, and F. Ardito. Lazy Rerouting for MPLS Traffic Engineering. Technical report, Università di Trento, Dipartimento di Informatica e Telecomunicazioni, March 2003. DIT-03-011.

[100] E. Salvadori, Z. Zsóka, and R. Lo Cigno. Dynamic Grooming in IP over WDM Networks: A Study with Realistic Traffic based on GANCLES Simulation Package. In *Proceedings of the 9th IFIP/IEEE ONDM*, Milan, Italy, February 2005.

[101] E. Salvadori, Z. Zsóka, R. Lo Cigno, and R. Battiti. A Framework for Dynamic Grooming in IPO Overlay Architectures. Submitted for publication, 2005.

[102] K. Sato, N. Yamanaka, Y. Takigawa, M. Koga, S. Okamoto, K. Shiomoto, E. Oki, and W. Imajuku. GMPLS-Based Photonic Multilayer Router (Hikari Router) Architecture: an Overview of Traffic Engineering and Signaling Technology. *IEEE Communications Magazine*, 40(3):96–101, March 2002.

[103] S. Sibal and A. De Simone. Controlling Alternate Routing in General-Mesh Packet Flow Networks. In *Proceedings of ACM SIGCOMM'94*, London, UK, August 1994.

[104] J. Skorin-Kapov and J. F. Labourdette. On minimum congestion routing in rearrangeable multihop lightwave networks. *Journal of Heuristics*, 1:129–145, 1995.

[105] N. Sreenath, B. H. Gurucharan, G. Mohan, and C. Siva Ram Murthy. A Two-Stage Approach for Virtual Topology Reconfiguration of WDM Optical Networks. *Optical Networks Magazine*, 2(3):58–71, 2001.

[106] R. Srinivasan and A. K. Somani. Dynamic Routing in WDM Grooming Networks. *Photonic Network Communications*, 5:123–135, March 2003.

[107] J. Stewart. *BGP4: Interdomain Routing in the Internet*. Addison-Wesley, 1998.

[108] J. Strand, A.L. Chiu, and R. Tkach. Issues for routing in the optical layer. *IEEE Communications Magazine*, pages 81–87, February 2001.

[109] S. Subramaniam, M. Azizoglu, and A. Somani. All-optical networks with sparse wavelength conversion. *IEEE/ACM Transactions on Networking*, 4(4):544–557, August 1996.

[110] H. Tangmunarunkit, J. Doyle, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Does AS size determine degree in AS topology? *ACM Computer Communication Review*, 31(5), October 2001.

[111] E.C. Tien and G. Mohan. Differentiated QoS routing in GMPLS-based IP/WDM networks. In *Proceedings of GLOBECOM*, volume 3, pages 2757–2761, 2002.

[112] B. Wang, X. Su, and C.P. Chen. A New Bandwidth Guaranteed Routing Algorithm for MPLS Traffic Engineering. In *Proceedings of ICC*, volume 2, pages 1001–1005, New York - USA, 2002.

[113] Z. Wang and J. Crowcroft. QoS Routing for Supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, 14(7):1228–1234, September 1996.

[114] J.Y. Wei. Advances in the Management and Control of Optical Internet. *IEEE Journal on Selected Areas in Communications*, 20(4):768 –785, May 2002.

[115] B. Wen, N.M. Bhide, R.K. Shenai, and K.M. Sivalingam. Optical Wavelength Division Multiplexing (WDM) Network Simulator (OWns): Architecture and Performance Studies. *Optical Networks Magazine Special Issue on "Simulation, CAD, and Measurement of Optical Networks"*, 2, September 2001.

[116] C. Xin, Y. Ye, T.S. Wang, and S. Dixit. On an IP-centric control plane. *IEEE Communications Magazine*, 39(9):88–93, 2001.

[117] Y. Xin, G. N. Rouskas, and H. G. Perros. On the design of MPλS networks. Technical Report TR-01-07. Technical report, North Carolina State University, Raleigh, NC, July 2001.

[118] Bülent Yener and Terrance E. Boult. A study of upper and lower bounds for minimum congestion routing in lightwave networks. In *Infocom 1994 Proceedings*, pages 138–149, 1994.

[119] J.Y. Youe and S.W. Seo. An algorithm for virtual topology design in WDM optical networks under physical constraints. In *Proceedings of ICC*, volume 3, pages 1719–1723, New Orleans, LA, June 1999.

[120] H. Zang, J. P. Jue, and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Optical Networks Magazine*, 1(1), January 2000.

[121] H. Zang, J.P. Jue, L. Sahasrabuddhe, R. Ramamurthy, and B. Mukherjee. Dynamic lightpath establishment in wavelength routed networks. *IEEE Communications Magazine*, 39(9):100–108, 2001.

[122] X. Zhang and C. Qiao. On scheduling all-to-all personalized connection and cost-effective designs in WDM rings. *IEEE/ACM Transactions on Networking*, 7(3):435–445, June 1999.

[123] X. Zhang, J. Y. Wei, and C. Qiao. Constrained multicast routing in WDM networks with sparse light splitting. *IEEE/OSA Journal of Lightwave Technology*, 18(12):1917–1927, December 2000.

[124] Z. Zhang and A. S. Acampora. A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength re-use. *IEEE/ACM Transactions on Networking*, 3(3):281–288, June 1995.

[125] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee. Dynamic traffic grooming in WDM mesh networks using a novel graph model. In *Proceedings of GLOBECOM*, volume 3, pages 2681–2685, 2002.

[126] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee. A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks. *IEEE/ACM Transactions on Networking*, 11(2):285 –299, April 2003.

[127] K. Zhu and B. Mukherjee. Traffic grooming in an optical WDM mesh network. *IEEE Journal on Selected Areas in Communications*, 20:122 –133, Jan 2002.

[128] K. Zhu and B. Mukherjee. A Review of Traffic Grooming in WDM Optical Networks: Architectures and Challenges. *Optical Networks Magazine*, 4(2):55–64, 2003.

[129] K. Zhu, H. Zang, and B. Mukherjee. A comprehensive study on next-generation optical grooming switches. *IEEE Journal on Selected Areas in Communications*, 21(7):1173–1186, September 2003.