ABSTRACT

HUANG, SHU. Network design and optimizations with applications in optical and wireless networks. (Under the direction of Dr. Rudra Dutta).

Typically, enterprise networks may incorporate Wide Area Networks (WANs), Metro Area Networks (MANs) and Local Area Networks (LANs). These networks have very different characteristics in terms of the physical media, data link layer and network layer protocols they use. In all these networks, well-designed network architectures are the key to achieving high performance at reasonable costs.

We study the optimization problems arised in the design of different networks, specifically, optical networks and wireless networks. In optical networks, we study the traffic grooming problem in Wavelength Division Multiplexing (WDM) networks with dynamic traffic demands. We present of detailed study of current research in this area and propose a new design problem for both single-link and multi-link networks. In wireless networks, we present new formulations for the design problem in Wireless Mesh Networks (WMNs) that take different inteference models into consideration and propose algorithmic methods to solve them.

Network design and optimizations with applications in optical and wireless networks

by

Shu Huang

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh

2006

Approved By:

Dr. Carla Savage	Dr. David Thuente
Dr. Rudra Dutta Chair of Advisory Committee	Dr. Harry Perros

To my parents and my brother . . .

Biography

Shu Huang was born on May 13, 1973 in ShangHai, China. He completed his undergraduate studies in electronic engineering at Beijing University of Posts and Telecommunications, China. He received a Bachelor of Science in 1995.

He spend the next several years in China Academy of Telecommunictions as an engineer with increasing level of experience and responsibility.

In 2002, he joined North Carolina State University to pursue his PhD's studies in Comupter Networking.

Acknowledgements

The process of finishing this dissertation would not have been possible without the guidance, support, coaching and mentoring of my advisor Dr. Rudra Dutta. I would like to express my gratitude to his sincere tremendous effort in guiding and correcting the writing of this dissertation.

I would also like to extend my appreciation to committee members, Dr. Harry Perros, Dr. David Thuente, Dr. Carla Savage.

Contents

List of Figures

List of Tables

Part I

Introduction

The design of networks plays an important role in the deployment of networks. Generally speaking, the network design is an optimization problem that, given various inputs such as the availability of network resources and traffic demands, outputs a solution such that some kind of objective function is optimized. The solution, i.e., the design of the network, has different content in different environment. For example, in optical networks, it can be the topology of the networks routing and wavelength assignment and traffic grooming. In wireless networks, it can be the network deployment, channel assignment, routing and scheduling, etc. Similarly, different networks may have very different constriants and objectives. For example, in optical networks, the number of wavelengths, the availability of wavelength converters are normally considered as expensive resources, thus impose important constraints on the design. It turns out that one of the well accepted objectives in optical network design is to minimize the cost of the network. However, in wireless networks, resources different from those in optical networks may be more constrained. For example, in Sensor Networks, because the batteries can not be recharged easily, the design of the network should keep the power consumption as low as possible. In addtion, based on the media sharing nature of wireless networking, the interference among different wireless attennas also imposes a vital constriant on the availability of resources. The objectives in wireless networks are usually the throughput, delay, etc.

Therefore, the design of networks provides insights into the following problems: 1) The allocation of network resources? 2) The analysis and effects of competitive and/or cooperative agents, protocols, protocols, network dynamics, network traffic and topology.

Part II

Optical Networks

Chapter 1

Wavelength Routed Networks

In this information age, we have witnessed the relentless demand for computer networks of higher bandwidth and lower costs. In order to fulfill this requirement, copper cables have been replaced by optic fibers in the core network for the much larger bandwidth optic fibers provide. It is well understood that optic fibers will eventually extend to individual homes (FTTH or Fiber To The Home) as a result of higher capacity demands of personal usage and lower costs of optic fibers and optical equipment. To fully utilize the large capacity of optic fibers, different modulation schemes have been proposed. Among them, a two-dimensional approach in both time domain and frequency domain multiplexing has been studied and widely accepted as likely to be dominant in the near future. The first generation of optical networks, those using optical fiber and communication at the physical layer, but not exploiting optical speed and reliability in switching or routing, have been deployed for some years. These networks typcially employ only time division multiplexing, and operate at typical speeds of 2.5 Gbit/s for the fiber. New optical technology has opened up the possibility of much higher bandwidths on a fiber, all the way up to the Tbit/s level. It is generally accepted that networks utilizing such technology and such bandwidth will prove indispensable for the commerce, business, education, and critical infrastructure of the coming decades, and will form the backbone of the next generation of wide area optical networks [2, 3, 4]. However, these advances also give rise to a more complicated network and many new problems previously not considered by network engineers.

With the advance of *Wavelength Division Multiplexing* (WDM) in optical networks, it is possible to multiplex several wavelength channels on a single optical fiber. The number of such wavelengths that can be multiplexed on a fiber now stands at about 200. Each wavelength on a fiber in such a WDM system can be viewed as a channel that provides an optical connection between two nodes. We call such a channel a *lightpath.* Traffic demands are aggregated into data frames by conventional TDM methods, which traverse lightpaths. Multiple lightpaths are now multiplexed on a fiber link using WDM. Lightpaths are originated from electronic traffic streams at the network node originating the traffic and terminated into electronic traffic streams at the node for which the traffic is destined by electro-optical Line Terminating Equipment (LTE), which are among the costliest of the network components used in such an architecture.

In the architecture of first generation optical networks, each wavelength channel would need to be electronically terminated at each intermediate node as well, and reconverted into an optical signal after being forwarded electronically. This so called OEO routing (Opto-Electro-Optic) would not only require many more costly LTEs, but interpose electronic processing and buffering in between optical transport segments, and hence increase delay, variability of delay, and the possibility of loss due to buffer overflow tremendously. However, advancement in the technology of *wavelength routing* gives us the capability of selectively bypassing some lightpaths optically, instead of converting every lightpath coming in on a single optical fiber to electronic signals. That is, a lightpath can traverse one or more intermediate nodes optically without the need of LTEs.

A lightpath thus becomes a more general transport mechanism for the next generation of optical networks, and it is possible for traffic demands at higher layers to be routed entirely on lightpaths, without reference to the physical fiber links in the network. The set of all such lightpaths therefore forms a topology which appears as the network topology to the higher layers. To distinguish it from the physical topology of the network formed by the fiber links, this topology has been variously called the *optical, logical,* or *virtual topology* [**?**, references thereof]. The problem of designing such virtual topologies may have different objectives, such as delay or hop count minimization, or throughput maximization. In recent times, it has become clear that the increasing bandwidth available even on a single wavelength will force successful network architectures to combine slower speed traffic streams with different source and destination nodes on a single lightpath using TDM techniques. In such a case, some OEO routing and LTE cost is unavoidable. The class of problems in which a virtual topology is designed and network traffic routed on the virtual topology with the objective of minimizing the electronic routing load on the network has been named *traffic grooming*.

Chapter 2

Dynamic Traffic Grooming

Computer and communication networking have been maturing over the past several decades, and has moved beyond the age of survival to the age of sophistication. The expectations of the end user from the network have also changed, and the concept of Quality of Service (QoS) and Service Level Agreements (SLA) have become pervasive. Until recently, it was assumed that such concerns were operative primarily in transport networks, that is at the highest level of aggregation of traffic in the planetary network hierarchy. At lower levels of aggregation, the network was seen to be composed of traffic networks, where QoS was neither feasible nor desired.

In this context, *traffic grooming* became an active area of research starting from the late 1990s. The new generation optical networks utilizing Wavelength Division Multiplexing (WDM) are currently in the process of being deployed to form the backbone networks of tomorrow. In WDM, multiple wavelength channels can be used over the same physical link of optical fiber using frequency multiplexing. Each wavelength channel can carry 10 Gbps with current technology, and higher rates are foreseen for the near future. Further, *wavelength routing* technology makes it possible to forward an optical signal at an intermediate node entirely at the optical plane, forming clear end-to-end optical channels that are called *lightpaths*. WDM networks utilizing wavelength routing can be modeled as multi-layer networks that consist of a virtual layer formed by such lightpaths implemented over a physical topology of optical fiber, and customer traffic routed at a second level, over the lightpaths of the virtual topology. The customer traffic demands are expected to be generally of much smaller bandwidth than the capacity of a single wavelength channel. Moreover, the traffic demands will be various different rates. For example, in generalized MPLS (GMPLS) [5, 6] networks, the traffic carried by this virtual layer are label switched paths, which can be of arbitrary bandwidth requirements. Because of the significant disparity between the the typical bandwidth request of a traffic component and the much higher capacity of a wavelength, it is well recognized that, to reduce the network cost, low speed traffic (referred to as subwavelength traffic) must be multiplexed (using Time Division Multiplexing) into lightpaths.

However, wavelength routing only allows the entire wavelength channel to be switched at the optical plane. If differentiated routing and forwarding of subwavelength traffic components contained in a wavelength channel is required, the optical signal must be terminated using Line Terminating Equipment (LTE), converted into digital eletronic signals, and input to an electronic logic device such as a traditional electronic router. At the end of the electonic routing operation, the packets must again be converted to optical signal and injected into outgoing lightpaths. This operations is called Opto-Electro-Optic (OEO) conversion, and is generally not desirable because it offsets the high speed and reliability of optical transport, and the OEO device is significantly more expensive than the optical switching equipment. Thus the subwavelength traffic must be packed into full wavelengths such that the cost of such OEO conversion may be optimized globally. This is the problem usually referred to as traffic grooming. The reader is referred to [4] for a survey.

In this literature, researchers have generally assumed that the magnitudes of traffic demands (given as a single traffic matrix) do not change with time. This assumption is reasonable for the following two reasons. First, in many core networks, low speed traffic requests are aggregated over several hierarchical levels of networks, and at many levels the bandwidth of the higher level network is sufficient to carry the aggregated flows from the tributary networks in terms of the average rates, but not the peak rates. Thus there is periodic buffer buildup and drainout, leading to some smoothing of traffic burstiness in such networks. Second, because of the importance of high speed traffic demands (in terms of the revenue the carrier will obtain), the network is designed such that the peak rates of traffic demands, which do not change drastically, are satisfied. Both reasons make the problem amenable to the static analysis.

However, recently the usefulness of the static approach has been seen as having clear limitations. As WDM optical networks are being deployed not only in Wide Area Networks (WAN) but also in Metropolitan Area Networks (MAN) and Local Area Networks (LAN), traffic demands have shown different dynamics. At the same time, the emergence of end-to-end QoS concerns has made it desirable to apply network design and resource provisioning techniques that were considered more suited to backbone networks to these lower level networks. In such networks, the magnitudes of traffic demands are more appropriately modeled as some functions of time. The traffic grooming problem has been generalized into this arena, giving rise to *dynamic* traffic grooming.

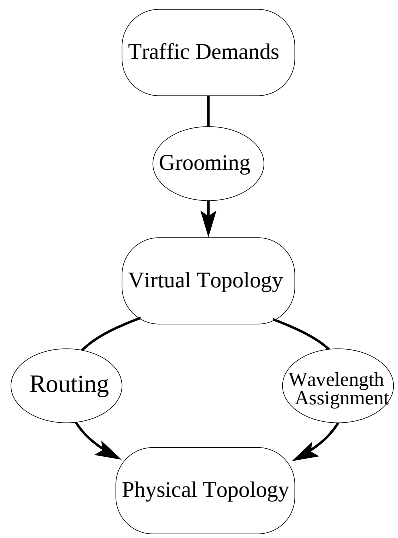


Figure 2.1: Dynamic traffic grooming subproblems

The static traffic grooming problem can be conceptually decomposed into three sub-problems: (i) the virtual topology design subproblem, (ii) the routing and wavelength assignment (RWA) subproblem, and (iii) the routing of traffic demands on the lightpaths, or grooming, subproblem. Fig. 2.1 shows the layered nature of these subproblems. Briefly, the network physical topology of optical fibers is an input to the problem, as is the set of traffic demands to be satisfied. The networks designer must decide what set of lightpaths to implement in the network; this is called the virtual topology subproblem. Having decided the virtual topology, the designer must specify a physical route for a lightpath from each source to each destination and assign to each lightpath a wavelenght out of a given set, such that no more than one lightpath of a given wavelength traverses each link, and the wavelength assigned to each lightpath is the same on all physical links. This is the

Routing and Wavelenth Assignment (RWA) problem, which has been extensively discussed and studied in optical networking literature; see [4] and references thereof for a detailed discussion. Finally, the subwavelength traffic demands must be routed over the lightpaths formed, so that each traffic demand is carried by a sequence of lightpaths that form a path in the virtual topology which carries the traffic from its source to its destination. Traffic is transferred from one lightpath to the next in the sequence by OEO routing. Global minimization of OEO routing or OEO equipment required at network nodes is often the goal of static traffic grooming, as mentioned above.

The dynamic traffic grooming problem can be understood in terms of exactly the same subproblems. However, the objective of grooming must be seen in a new light. Unlike the static problem, in the dynamic traffic grooming problem the solutions to these subproblems need to satisfy time-varying traffic. Thus the solution itself must vary with time. At the least, the mapping of traffic demands to the virtual topology must change. Also, network designers can take the advantage offered by reconfigurable optical switches to dynamically adjust the virtual topology in response to traffic demand changes, in that case the RWA must also be readjusted to map the changed virtual topology onto the unchanging physical topology.

It is important to note that the focus of grooming traffic shifts as a consequence of the above. Reduction of OEO costs may continue to be an objective of traffic grooming. But the primary objective may now well be a minimization of the blocking behavior of the network; this is not particularly relevant in static traffic grooming because with good planning the entire traffic matrix is expected to be carried by the network, but making a similar 100% guarantee under statistically described dynamic traffic may be prohibitive in cost and not desirable. Similarly, the consideration of fairness is not relevant for the static problem, but may become an important one for the dynamic case.

Another change of focus relates to the complexity of the grooming solution. In static grooming, solution approaches of significant computational complexity may be practical, since such solutions are expected to be computed off-line, with a given estimate of traffic that is expected to be valid for a reasonably long time. For the dynamic case, the solution will be computed on-line, and re-computed over normal network time scales. Thus it is essential that the algorithms to compute new solutions be of low computational complexity. Similarly, an algorithm that can be computed in a distributed manner is likely to be of far more practical use in the dynamic context than one that requires a centralized approach;

this distinction is less significant in the static case. Thus in various ways, the goal and priorities of grooming changes in the dynamic traffic context, and this is what we refer to as the changing role of traffic grooming. Finally, as the field evolves, it is likely to come to be perceived as a general class of network design problems where the cost component is largely concentrated into specialized network node equipment that will enter the mainstream in the future, such as optical drop-and-continue, wavelength converters, or OTDM switches.

The connection with the work in the Internet Engineering Task Force (IETF) in the GMPLS context is worth remarking upon. The original definition of Multi-Protocol Label Switching (MPLS) in the Networking Working Group of the IETF, building on earlier paradigms of tag switching and cut-through switching, was motivated by the need to reduce the forwarding burden on core routers. In label switching, an additional header is attached to Internet packets that carry information regarding flows to which each packet belongs. Once a flow, called a Label Switched Path (LSP) in MPLS, is set up, a Label Switching Router (LSR) in the path can forward packets bearing the label corresponding to the flow with much less processing than for a normal packet. In Generalized MPLS (GMPLS) [5, 6], time slot positions for TDM transport and wavelength channels for optical transport can also act as labels. It was soon realized by the networking community that label switch routing could also serve as an enabling mechanism for traffic engineering (TE), and flow-level QoS, because it allowed the identification of flows to routers. There has been significant recent work in defining extensions and signaling for the interaction of GMPLS and underlying networking layers, including SONET and other optical transports, and the communication of traffic engineering information between underlying networks and GMPLS [7, 8, 9].

However, these developments have focused (as appropriate for the role of the IETF) on enabling technology rather than design strategies. In keeping with the original guiding principles of the Internet, the network administrator is provided mechanisms to set up TE or QoS actions; but what actions are to be taken is left up to the administrator, who must look elsewhere for algorithms that provide policy or strategy decisions. To put it simply, all the mechanisms to set up LSPs is provided, but what LSPs to set up must be decided by the network administrator or operator. It is in this sense that research work such as traffic grooming provides a necessary complement to the development of enabling technology.

Because of the wide deployment of WDM networks, efficient operation under dynamic traffic is an area of practical interest to service providers. Efforts at different layers have already started in the arena of enabling technology to make the network friendly to dynamic traffic. At the lower layer, in the legacy Synchronous Optical Network (SONET) or the Synchronous Digital Hierarchy (SDH) networks, the hierarchical rates defined for multiplexing/demultipexing make it inefficient to carry dynamic traffic requests. To overcome this intrinsic inefficiency, two mechanisms, Virtual Concatenation (VCAT) (as defined by the International Telecommunication Union in its recommendation [ITU-T G.707]) and the Link Capacity Adjustment Scheme (LCAS) (as defined in [ITU-T G.7042]) have been developed for Next Generation SONET. At the higher layer, part of the motivation to generalize MPLS to GMPLS has been to provide a uniform control plane to LSRs that operate at IP/MPLS level as well as network equipment that operate at fiber, wavelength and circuit level. Dynamic traffic grooming is thus a timely and emerging research area. Our focus in this survey is this research area, which is expected to provide algorithms that supply designs or policies for network operation.

While a significant number of studies have appeared recently on dynamic traffic grooming, there is as yet no single resource that provides a comprehensive introduction to the problem as well as to the literature. In this chapter, we hope to fill this void by providing an insight into the factors that must be considered in formulating a dynamic traffic grooming problem, and presenting a survey of the literature.

The rest of the chapter is organized as follows. In Section 2.1, we briefly discuss network node architectures for traffic grooming networks, because it is an important factor in dictating the goals of the network design problem. We provide discussion regarding the formulation of the dynamic traffic grooming problem either as a resource allocation problem or a policy design problem in Section 2.2. This also allows us to present a classification of the literature in Section 2.3, followed by a detailed literature survey according to our classification. We conclude with a few remarks on future directions in Section **??**.

2.1 Node Architectures

The extent to which subwavelength traffic components may be manipulated (and thus what grooming actions may be performed) is determined by the network equipment that are available at the nodes. Accordingly, in this section, we provide a brief overview of nodal capabilities. A more detailed discussion, with some discussion of future switch capabilities, may be found in [10].

Generally speaking, the traffic entering/leaving a node equipment can be described by a tuple (optical fiber, wavelength, time-slot). Thus, a "perfect" switching node would perform a complete permutation, *i.e.* the traffic from any fiber, any wavelength, and any time slot would be possible to switch to any other fiber, wavelength, time slot. However, due to considerations of cost and scalability, different node architectures are deployed in reality that have less than perfect switching capability. These impose different constraints on the grooming problem. We will show in Section 2.2.3 how a mathematical formulation for the dynamic traffic grooming problem requires careful examination of the node architectures. A generic modeling of the constraints that applies to different architectures is also an interesting problem.

The basic conceptual building blocks of such switches can be broadly divided into optical components, which manipulate optical signals, and thus operate at the level of entire wavelength channels, and electronic or digital components, which are capable of manipulating individual bytes and packets as electronic signals, as in traditional routers and electronic computers. Optical networking switches will in general have some of each type of component, and can be characterized by the capabilities of each of these. When a number of signals are multiplexed into a carrier, multiplexers (MUX) and de-multiplexers (DEMUX) are required at the sender and receiver respectively. If an equipment has the capability to de-multiplex signals, then selectively switch some of them to another switching equipment at the same node, while passing others through to a multiplexer for outgoing signals, it is called an *Add-Drop-Multiplexer* (ADM). Such an equipment performs only one decision for each de-multiplexed flow (whether to drop it or to pass it through). If, in addition, the equipment has the capability to choose which of several outgoing ports a signal is passed through to, it is called a *Cross-connect* (XC).

SONET/SDH ring networks were one of the first optical networking architectures to be used in practice, and continue to be important today. In SONET rings, only one optical channel on each fiber is used. Fibers are usually interconnected by *SONET Add-Drop-Multiplexers* (SADMs), which are digital equipment that have the capability to switch traffic at time-slot level. Thus the MUX/DEMUX refers to individual traffic streams time-division multiplexed in the optical signal. At a ring node, there is only one other node from which an incoming link exists, and only one other node to which an outgoing link exists. Thus Add-Drop functionality is all that is required. In SONET mesh networks, fibers are interconnected by *Digital Cross-Connects* (DXCs or DCSs), which, unlike ADMs, handle multiple input and output fiber ports. DXCs, which perform switching at time-slot level, can be characterized by p/q, where p represents the port bit rate and q represents the bit rate that is switched as an entity. For a comprehensive description of SONET/SDH, see [11].

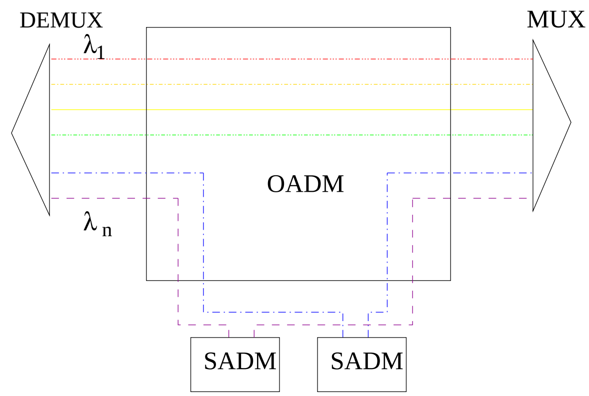


Figure 2.2: An OADM Architecture

For WDM networks, multiple wavelength channels are frequency multiplexed in each fiber links, and lower rate traffic streams are time division multiplexed in each wavelength channel. The digital equipment at the node can perform switching actions on the lower rate traffic streams by utilizing Synchronous Transport Signal (STS) in the optical signal. In WDM ring networks, an Add-Drop method as above can be used, but now *Optical ADMs* (OADMs) are used to selectively by-pass some wavelengths along the ring, while others are dropped into digital equipment, which may be SADMs. This forms the simplest node structure that can be used in optical grooming networks, and is shown in Fig. 2.2. The various wavelength channels frequency multiplexed in the fiber are represented by $\lambda_1 \ldots \lambda_n$. The by-passing of wavelength channels creates *lightpaths*, channels that are optically continuous over multiple physical fiber links. In Fig. 2.2, the first four wavelengths are by-passed in this fashion, whereas the last two are dropped (and added, at the output). It is possible to re-generate a lightpath signal on a different wavelength entirely by optical hardware (without converting the signal into the digital electronic plane), this is called wavelength conversion. However, such equipment is quite costly, and in many cases practical node architectures may not include such converters. Without wavelength conversion capability, lightpaths must obey the *wavelength-continuity constraint*, *i.e.* a lightpath must be assigned the same wavelength on the fiber links it traverses. For each added/dropped wavelength, an SADM is dedicated to process the traffic the wavelength carries electronically. The number of SADMs at a node determine the number of wavelength channels for which traffic can be switched at the timeslot level, thus this number characterizes in part the switching power of the node. It is well recognized that the cost of transceivers is the main contributor to the network cost, therefore the number of SADMs available at an OADM is usually either the objective to minimize, or a constraint to which the optimization problem is subject. This problem is referred to as ADM constrained grooming in [12]. Furthermore, if the SADMs on the different wavelengths are isolated (as shown in Fig. 2.2), not only lightpaths but traffic components need to obey the wavelength-continuity constraint because the traffic dropped at a wavelength has to be sent onto the same wavelength in order to be forwarded to its destination, as in [12]. This constraint can be relaxed if a digital switching fabric is available such that the traffic added/dropped by SADMs can be reshuffled and re-injected into other SADMs, resulting in a more powerful switching node. Fig. 2.3 shows an example of such a node, with optical MUX/DEMUX and OADM, and SADMs on each dropped wavelength connected by a DXC.

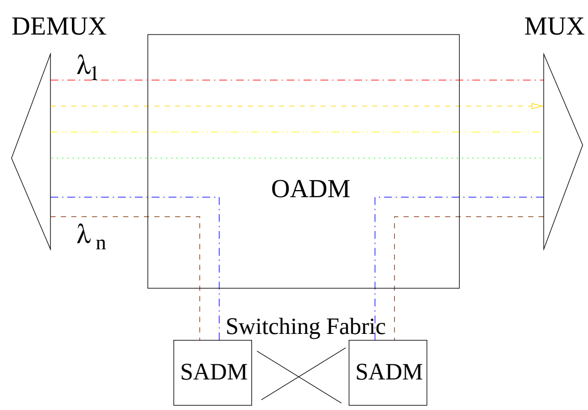


Figure 2.3: An OADM architecture that allows cross-connect of local traffic

In contrast to OADMs, which usually have predetermined add/drop wavelengths, *Reconfigurable OADMS* (ROADMs) allow a network administrator or operator to dynamically select what wavelengths to drop or by-pass. The reconfigurability does not represent an increase in the power of the switch in terms of how much traffic can be switched, but introduces more flexibility. The number of maximum wavelengths that can be dropped

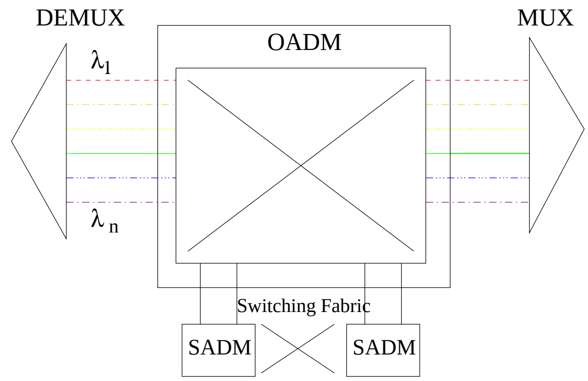


Figure 2.4: An ROADM architecture

characterizes the power of the switch, as well as the digital switching capability (as before). For a comparison of different ROADM architectures, refer to [13]. An example is shown in Fig. 2.4.

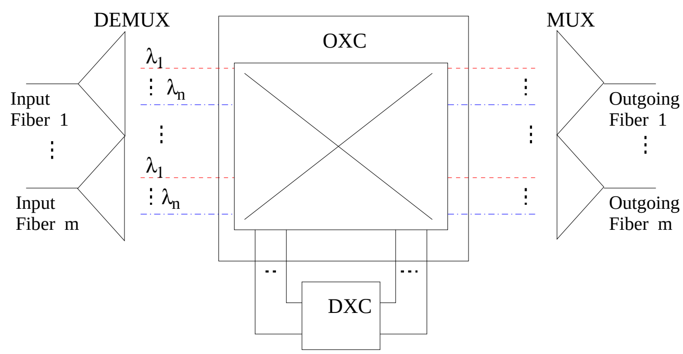


Figure 2.5: An OXC architecture that has grooming capability

In all the above, the optical part of the switch is only an ADM, and the electronic part is an ADM or an XC. These can all be viewed as a special case of *Optical Cross-Connects* (OXCs), the most general class of grooming switches, which are widely expected to be deployed in realistic mesh topologies. In such switches, the optical ADM is replaced by an optical XC. Thus wavelength channels can not only be by-passed to form ligthpaths, but these lightpaths can be switched to specific output ports. An OXC is similar to an ROADM, but can accommodate incoming fibers from multiple nodes, similarly outgoing fibers to multiple nodes. Three broad classes of OXCs have been defined (refer to Telcordia's

Optical Cross-connect Generic Requirements GR-3009-CORE):

- *Fiber switch cross-connect* : the entire signal carried by an incoming fiber is switched to an outgoing fiber, cannot perform different actions for different wavelength channels of timeslots.
- *Wavelength Selective Cross-connect* : can switch a subset of the wavelengths from an input fiber to an output fiber, obeying wavelength continuity constraint.
- *Wavelength Interchanging Cross-connect* : WSXC with wavelength conversion capability.

In addition, time slot multiplexing/demultiplexing and grooming can be performed by a DXC if it is incorporated in the node. Fig. 2.5 shows an example of OXC that has grooming capability, with m input and m output fiber ports. (Usually, the number of input fiber ports is equal to the number of output fiber ports; however, in [14], the design of strictly non-blocking OXCs with different numbers of input and output fibers has been studied.) An OXC usually has two separate switching fabrics, the wavelength switching fabric that switches traffic at the wavelength level, and the grooming fabric that switches traffic at the time-slot level [15]. Since the grooming fabric can be viewed as a DXC, to avoid confusion, the cost is usually modeled in terms of the number of transceivers, instead of SADMs as in SONET ring networks. Note that both the transceiver and the SADM can be seen as terminating a lightpath into digital equipment; thus this cost measure can be generalized as the number of LTEs required.

Other node capabilities related to the ones described above are possible. A node intermediate in power between an ROADM and an OXC called *Optical Add-Drop Switch* (OADX) has also been defined and is commercially available; however, we do not discuss it here because from the grooming point of view such as node is equivalent either to an ROADM or an OXC. In [16], a node is modeled as trunk-switched and a generalized framework for analyzing *Trunk-Switched Networks* is addressed. The authors introduce the concepts of trunks and channels, whose definitions are node architecture dependent. Trunks can be viewed as forming a virtual layer, and an input channel can be switched to any output channel at a full-permutation node, as long as both channels are within the same trunk. For instance, without wavelength converters, a wavelength can be viewed as a trunk and if time-slot switching is permitted, a time-slot can be viewed as a channel. In [17], by the same authors, a network with heterogeneous node architecture is studied. However, this framework does not address the case of a node which combines different node architectures. For example, while the added/dropped wavelengths can interchange time slots through the switching fabric, an OADM has some bypassing wavelengths (trunks) in which time slots can not be switched.

As the above discussion shows, depending on the node architecture, a node can operate at the fiber, wavelength, or time-slot level, and at each level, it may have full or limited functionality. In addition, some variants are worth mentioning. For instance, to avoid the cost of full grooming DXCs, the grooming functionality can be separated into two levels, where the higher level is a coarse groomer that deals with high speed traffic streams and the lower level is a finer groomer that deals with low speed traffic streams. The authors of [18] consider such a situation, and remark that using the proposed mixed-groomer node architecture is beneficial in terms of reducing both the switching cost and the number of wavelengths required. In the node architecture introduced in [19], an extra waveband layer is inserted between the wavelength and the fiber layer. In [20], the authors describe a node architecture, Multicast-Capable Grooming Optical Cross-connect. Using the embedded strictly non-blocking splitter-and-delivery (SaD) switches, the proposed dynamic tree grooming algorithm can be supported. In [21], another Multicast-Capable Optical-Grooming Switch architecture is introduced. Instead of using SaD switches, it has two stages of optical switching. Multicast traffic leaving the first stage optical switch is sent to a splitter bank and then switched by the second stage optical switch.

2.2 The Dynamic Traffic Grooming Problem

In this section, we make some general observations to indicate the scope of problems dealt with in literature that we consider as coming under the umbrella of dynamic traffic grooming. Broadly, we include both problems that take an essentially dynamic approach to changing traffic and problems that convert this changing nature into a static design problem. However, the underlying problem should be motivated by the changing nature of traffic. Also, we consider the study to come under grooming only if the multiplexing of subwavelength traffic is considered to contribute to the cost model or constraints in some manner. We exclude literature from our scope if the only consequence of subwavelength traffic is seen to be the required multiplexing, because such studies are more appropriately considered to fall under the more established research areas of routing design and resource allocation with multiplexing. These considerations prompt us to consider out of scope studies such as [22], which is in effect static grooming study; or [23], which is more appropriately considered a restoration strategy design at the lightpath level. We use the concepts developed in this section to present a categorization of the literature and a detailed survey in Section 2.3.

2.2.1 Design and Analysis Problems

In [24], we classified the dynamic traffic grooming problem into two broad categories: the *design problem* and the *analysis problem*. The distinction, while not an absolute, is a practically useful one in understanding approaches to the problem and categorizing them.

- The network *design* problem focuses on the state space; a time-varying one for the dynamic problem. Given a model of behavior of the network and some quantities of interest to optimize, the design problem attempts to find optimal settings of controllable parameters.
- The network *analysis* problem focuses on modeling the behavior. Given an *a priori* policy of network control under dynamic traffic events, such as arrival, departure, increment, decrement; the analysis problem attempts to develop a predictive model of some quantities of interest, under changing values of input parameters, such as arrival rates.

The two problems are complementary, because the design problem presupposes a model that allows computation of the goal under specific resource allocation and policy, and the analysis problem presupposes an existing policy and resource allocation under given traffic conditions. In the area of dynamic traffic grooming, analysis problems considered in literature generally address the blocking performance of the network under some given grooming policy, as experienced by arriving subwavelength traffic components. The design problems considered in literature show a larger variety both in the problems formulated as well as the approaches taken, and we discuss more of them in the rest of this section.

At the end of this section, in Table 2, we use the distinction between design and analysis problems as our first categorization of literature on the dynamic traffic grooming problem. In Section 2.3, we include surveys of both categories of literature.

2.2.2 Quantities of Interest in Design

We briefly list the basic quantities in terms of which the design problem is defined, with accompanying notation.

- Let N be the set of nodes and A be the set of directed fiber links in the physical topology graph. We assume that the physical topology does not change with time.

- Let S be the set of traffic demands denoted by the source-to-destination node pairs in the network; S may consist of all distinct ordered pair of nodes, but may also be a subset of it because some node pairs do not have traffic between them.

- Let $\Lambda_{|N|\times|S|} = [\lambda_n^s(t)]$ be the traffic matrix, where $\lambda_n^s(t)$ is the time varying traffic flow for the node-flow pair (n, s). Specifically, $\lambda_n^s(t)$ is λ_s, if at time t, the traffic demand s is sourced from node n, and has magnitude λ_s. Similarly, $\lambda_n^s(t)$ is $-\lambda_s$, if the traffic demand s is destined to node n, and 0 if n is neither the source not the destination of the traffic demand s. We suppose λ_s for every s is in units of a basic rate, and the capacity of a wavelength is C, in the same units.

- Let the number of wavelength channels available on each physical fiber link by W, wavelengths are numbered from 1 to W on each fiber.

- Let the matrix of the physical topology be $P_{|N|\times|A|} = [p_n^{(a)}]$, where $p_n^{(a)}$ is 1 if the fiber a is sourced from node n, -1 if it is destined to n, 0 otherwise.

- Let L be the set of lightpaths, and let $V_{|N|\times|L|\times W} = [v_{n,w}^{(l)}(t)]$ be the matrix of wavelength layered virtual topology, where $v_{n,w}^{(l)}(t)$ is 1 if at time t, lightpath l is sourced from node n and uses wavelength w, -1 if it is destined to n and uses wavelength w, 0 otherwise.

- Let $R_{|A|\times|L|\times W} = [r_{a,w}^{(l)}(t)]$ represent how the virtual topology is routed on the physical topology and assigned wavelengths, where $r_{a,w}^{(l)}(t)$ is 1 if lightpath l uses the wavelength w on fiber link a at time t, 0 otherwise.

- Let $G_{|L|\times|S|} = [g_l^{(s)}(t)]$ represent how the traffic demands are routed on the virtual topology, where $g_l^{(s)}(t)$ is λ_s if the traffic demand s traverses lightpath l at time t, 0 otherwise. This represents the case that traffic bifurcation is not allowed; additional variables can be introduced to represent bifurcated or diverse routing of traffic demands.

In general terms, the *input* to the dynamic traffic grooming problem are:

(i) the traffic demand matrix Λ, a function of time,

(ii) the resource availability (includes physical topology P, number of wavelength channels W, etc.), generally not varying with time, and

(iii) the node architecture (limits to grooming capability, etc.), also generally not varying with time.

The *output* of the dynamic traffic grooming problem are:

(i) the virtual topology V,

(ii) the routing and wavelength assignment V for the virtual topology on the physical topology P, and

(iii) the routing G of the traffic demands on the lightpaths of the virtual topology.

In general, all of the outputs are functions of time.

2.2.3 Basic Constraints

Constraints on the node architecture

- As we observed in Section 2.1, the total OEO processing capability of a node is directly constrained by the finite number of LTEs (SADMs or transceivers) at the node. This is expressed as:

$$\max\left(\sum_{l:v_{n,w}^{(l)}>0}\sum_{w} v_{n,w}^{(l)}(t),\ \sum_{l:v_{n,w}^{(l)}<0}\sum_{w} -v_{n,w}^{(l)}(t)\right) \leq \text{LTE}_n \quad \forall n \tag{2.1}$$

where LTE_n is the number of LTEs available at node n.

- In Section 2.1, we have shown that different node architectures may also result in different constraints on the feasible grooming solutions. For instance, the unavailability of wavelength converters imposes the wavelength continuity constraint in the RWA problem. Because the wavelength converters are expensive, most researchers assume that they are absent in the network. Consequently, lightpaths must obey the wavelength continuity constraint. That is;

  $v_{n,w}^{(l)}(t)$ is 1 if at time t lightpath l is sourced from node n, -1 if at time t lightpath l is destined to node n, 0 otherwise.

  Depending on the node architecture, there may be further constraints on the set of wavelengths a local transmitter can be tuned to. For example, practically, transmitters may be equipped with lasers with limited tunability (*e.g.*, a recent OADM card provided by a major vendor can only be tuned to a band that has two predetermined wavelengths). However, if the wavelengths that are dropped/added are reconfigurable and completely selective, such a constraint is not required.

Constraints on the RWA problem

To ensure correct RWA, we can use the following constraint or similar:

$$P_{|N|\times|A|}R_{|A|\times|L|\times W} = V_{|N|\times|L|\times W} \tag{2.2}$$

To ensure one wavelength on a fiber is assigned to at most one lightpath, we can use:

$$\sum_{l} r_{a,w}^{(l)}(t) \leq 1 \quad \forall a, w \tag{2.3}$$

Constraints on the traffic routing

We use $V_{|N|\times|L|} = [v_n^{(l)}(t)]$ to denote the virtual topology at time t. $v_n^{(l)}(t)$ is 1 if the lightpath l is sourced from node n at time t, -1 if the lightpath l is destined to node n, 0 otherwise. Note that the virtual topology is the sum of the wavelength layered virtual topology, that is:

$$V_{|N|\times|L|} = \sum_{w} V_{|N|\times|L|\times W} \tag{2.4}$$

The following constraint ensures the traffic demands are properly routed on the virtual topology.

$$V_{|N|\times|L|}G_{|L|\times|S|} = \Lambda_{|N|\times|S|} \tag{2.5}$$

To ensure the capacity of a lightpath is obeyed, we have:

$$\sum_{s} g_l^{(s)}(t) \leq C \quad \forall l \tag{2.6}$$

2.2.4 Static and Dynamic Formulations of Design

Static Formulation - Resource Allocation

While traffic demands change with time, the change may be partly or wholly *predictable*. As an extreme case, the nature of variation of traffic with time may be completely deterministic. If the value of the traffic demands at all times (over a period of interest) is known with certainty beforehand, the problem can be seen as some variation of a general *resource allocation* problem, and a static formulation of the problem is most appropriate.

In this model, the traffic is deterministically given over some period of interest, possibly as a sequence of traffic matrices, $\Lambda(t_0), \ldots, \Lambda(t_n)$. The period may be infinite, by specifying that the pattern of traffic matrices repeats; this is essentially a scheduling problem. This model is amenable to an Integer Linear Program (ILP) formulation [25]. One obvious approach to such a problem is to eliminate the effects of time-variation altogether by simply designing for the peak values each traffic component assumes in the entire set of matrices. However, as shown in [26, 27], using the traffic matrix formed by the peak rates may result in requiring an unnecessarily large amount of resources. The reason is the space-time nature of the dynamic traffic grooming problem, which is left out of consideration in this approach. The traffic matrix of peak rates is an overestimation of the traffic demands, because the dynamic nature of traffic spreads peak rates out along the time dimension. Thus this problem, while a static problem, is distinct from the static grooming problem.

Dynamic Formulation - Policy Design

On the other hand, unpredictability or uncertainty may be seen as an essential characteristic of the traffic model. In such cases, the dynamic nature of the problem needs to be explicit in the problem formulation. The problem must be seen as one of supplying a *policy design* for the network, that is an algorithm that the network control plane can employ to make decisions in response to traffic change events, with state and action space defined as follows:

State space: Since traffic events can occur and network actions can be taken only at discrete points in time, we represent $\Lambda(t)$ as a discrete-time temporal process, Λ_i is the traffic matrix at time epoch t_i (a time epoch is defined as an instant at which a dynamic traffic event occurs). Then, each Λ_i is associated with a virtual topology V_i, a routing and wavelength assignment R_i, and a traffic routing G_i. The tuple $\{V_i, R_i, G_i\}$ is referred to as the grooming solution at time t_i. Then, the network state at time t_i can be described by the tuple $\{\Lambda_i, V_i, R_i, G_i\}$.

Action space: According to the layer it will affect, the actions taken by the network control algorithm can be classified as follows:

- Call Admission Control (CAC) actions, where two possible actions are **reject** and **accept**. If a traffic change is accepted, actions on other layers may follow. Note that while we use the term "call", the events may be more general ones than arrivals of entire subwavelength traffic demands; for example it may be an increment or decrement to the magnitude of a traffic connection already established. However, the network action must still start with a decision regarding whether to accept or reject the increment.

- Network layer routing actions. Once a change is accepted, the changed traffic will be either routed on the existing virtual topology, or it will trigger virtual layer actions. The actual route of the subwavelength call on the virtual topology must also be determined according to some policy. When the change is in the nature of a traffic decrease, network layer action may also be triggered to rearrange the routing of remaining traffic, see below.

- Virtual layer setup, teardown, or routing actions. To route the changed traffic component, new lightpaths may be set up. These may be either a direct lightpath, or a combination of new lightpaths, which may be further utilized in conjunction with existing lightpaths to route the changed traffic component. For new lightpaths, routing and wavelength assignment is performed. Similarly when traffic decreases, lightpaths may be also torn down in response.

- Re-routing Actions. Furthermore, if disruption of existing traffic is allowed, the actions may include rerouting (or even terminating) some existing traffic. Existing subwavelength traffic may be rerouted on the virtual topology, or existing lightpaths may be rerouted on the physical topology.

For each action, $\{V_i, R_i, G_i\}$ will change to $\{V_j, R_j, G_j\}$. The goal of the policy will be always to maximize some reward function, akin to the objective function for a static formulation; we discuss some possible goals later in this section.

Referring back to our discussion regarding Fig. 2.1, we see that the physical topology at the lowest layer does not change with time, and the highest layer, the traffic demands to be carried, do change with time. Thus dynamic traffic grooming strategies can be seen as the algorithms executed by the network to perform a time-varying mapping of the traffic onto the network resources, using the routing, wavelength assignment, and grooming, to satisfy the demands and satisfy some goal of network operation such as operating cost minimization or maximization of utilization.

2.2.5 Models of Non-deterministic Traffic Variation

For the dynamic formulation, traffic variations are not wholly predictable, but the time-variation of traffic may nevertheless be modeled or characterized to some extent. Different models can be designed to reflect realistic network conditions, we list a few below.

- $\Lambda(t)$ is a Poisson process, and the model is simply one of subwavelength traffic component arrival/departure. In the general context of dynamic traffic grooming, it is reasonable to assume that $|\Lambda(t) - \Lambda(t + \Delta t)|$ is small for a short time period Δt, which motivates this model.

- Traffic demands are preferred to be serviced within time windows [28]. This is a generalization of the simple arrival-departure model. Instead of each traffic component requiring to be serviced at the instant (or as soon after as possible) that it arrives, every traffic component specifies a window of time within which the traffic component must be carried. The arrival process may again be Poisson, or some other process.

- Traffic demands are restricted by specified bounds. Such bounds may be provided by the traffic components themselves, or they may be imposed by the available resources. For example, the number of SADMs available at the node (referred to as t-allowable traffic in [26]). Let SADM_n be the number of SADMs at node n, then the traffic matrices must satisfy:

$$\max\left(\sum_{s:\lambda_n^{(s)}>0} \lambda_n^{(s)}(t), \sum_{s:\lambda_n^{(s)}<0} -\lambda_n^{(s)}(t)\right) \leq \text{SADM}_n \cdot C \quad \forall n$$

- Traffic components change in magnitude over time in *increments* and *decrements*. The process by which increments and decrements occur may be Poisson or some other.

- Entire traffic matrices are specified as in the deterministic model, but the time epochs t_i are not deterministic, and varies according to some random process.

2.2.6 Design Goals

The goal of either resource allocation or policy design is to minimize some measure of cost in provisioning and operating the network, and/or to maximize the benefit from the network. This can be embedded as cost function(s) in a static formulation, and reward function(s) in a decision formulation. In the literature, different goals have been articulated, some representative ones include:

- Minimize the network cost; these are more suitable for the static, resource allocation, view:
  - Number of ports at network nodes (converters, LTEs, wavelengths).
  - Amount of OEO processing.
- Maximize the revenue by providing better service or better utilization of the network resource; more appropriate for the dynamic, policy design, view:
  - Minimize the blocking probability.

- Minimize the provisioning time (time to setup a connection for an arrival, traffic delay, etc.)
- Minimize the disruption to traffic already being carried.
- Minimize the unfairness (*e.g.*, traffic demands having different bandwidth requests should have approximately the same blocking probability).

These goals are usually correlated in a way that makes it impossible for them to be optimized simultaneously. Therefore, some kind of trade-off or preference must be considered. For example, in [29], the network architectures for WDM SONET rings that have the minimal SADM cost are studied, but subject to a limited number of wavelengths. In [25], an MILP for the dynamic traffic grooming problem with the objective of minimizing the SADM cost is solved by two phases, where, in the first phase, the number of wavelengths is minimized. In [30], the authors propose a connection admission control mechanism that provides good fairness without over-penalizing the overall blocking probability. In [26, 31], the objective is to design networks with the minimal SADM costs while keeping the existing traffic undisrupted (non-blocking in the strict sense).

2.3 Survey of Literature

2.3.1 Literature Classification

Based on the observations we made in the last section, we present an organized view of the literature on the dynamic traffic grooming problem in Table 2. Because all the categories are not orthogonal, several papers appear in multiple places in this table. Thus this table should be thought of as an organization, even more than a categorization.

Moreover, some studies address more than one category of problem. For example, consider the variants of blocking probability that are considered in the literature. The blocking characteristic of a network can be classified as strict-sense non-blocking, wide-sense non-blocking and rearrangeable non-blocking (*e.g.* in [26]). If the network resources can guarantee strict-sense non-blocking, then all the new arrivals will be satisfied, and the policy design problem may not be addressed since it is trivial. However, if network cost considerations dictate accepting lesser blocking performances, to design a wide-sense non-blocking or rearrangeable non-blocking network, both the problems of resource design and policy design (to route new arrivals) are likely to be addressed.

In the remainder of this section, we go on to survey the literature in more detail. Table 3 contains the list of all papers surveyed below, with short descriptions that, combined with Table 2, indicate where the detailed survey of each paper appears below. A fuller description of some of this literature may be found in [32].

2.3.2 Analysis Studies

As we have discussed in the previous sections, the resource and policy design problems are in essence optimization problems. In order to evaluate the performance (usually, the blocking probability) of a design, practitioners often resort to massive simulations. As simulation results are generally specific to the input (arrival and departure rates, etc.) and time consuming, analytical modeling is not only interesting in its own right but practically meaningful. In literature, the metric of greatest interest is the blocking probability, i.e., the ratio of the number of accepted arrivals to the total number of arrivals. In order to accept an arrival, the subproblems described in Fig. 2.1 should be solved. We distinguish two cases, the single-hop case and the multi-hop case (referred to as dedicated-wavelength TDM and shared-wavelength TDM in [16]). In the former case, a new arrival is accepted if it can be routed on a single lightpath (either an existing one or a new one to be established) from source to destination. In the latter case, the arrival is allowed to traverse multiple lightpaths, which could be a combination of existing lightpaths and newly established lightpaths. In addition, some routing and wavelength assignment algorithms must be assumed, e.g., the shortest path routing and random wavelength assignment algorithms considered in [16, 17]. In queuing networks, we also distinguish single-rate and multi-rate requests. In the single-rate model, all traffic demands have the same magnitude. The model simplifies the analysis significantly. However, in grooming networks, the multi-rate model may be more realistic because traffic demands are usually subwavelength, thus in units of some basic rates (say, OC-3).

Another difficulty comes from the traffic model. It is well known that the Poisson model fails to capture the self-similarity of the traffic pattern in networks. In addition, in grooming networks, traffic demands usually traverse multiple physical/logical hops. Therefore, the link load correlation becomes an important issue.

All these challenges and difficulites make the exact queuing analysis intractable. Accordingly, researchers have made different assumptions and simplifications. In the following section, we survey related works in this field.

Multihop models with correlation

In this section we describe those studies which make the more general assumption that end-to-end traffic components may be carried either on a single lightpath from source to destination, or on a sequence of lightpath. Additionally, these studies attempt to represent the correlation between the link loads in some manner.

As previously mentioned in Section 2.1, in [16], Srinivasan et al. had presented a framework for analyzing the performance of Time-Space Switched optical networks. In [17], this framework is applied to networks with heterogeneous node architectures. Assuming a single rate model, the blocking probability for a path is computed recursively from a two-hop path model. The authors assume *Markovian correlation*, i.e., the traffic on a link only depends on its previous link. In [37], the authors extend the work to the multi-rate case.

Washington et al. also study the mulithop model with correlations [34]. ithey consider the blocking probability on tandem networks, i.e., a unidirectional path virtual topology. The authors consider the multi-rate arrival model on existing lightpaths. A path network is first decomposed into subsystems consisting of two adjacent nodes and analyzed exactly by a modification of Courtois' method. The first step of the Courtois' method that requires solving a system of equations is replaced by solving a multi-rate model for the *exact* conditional steady-state probabilities. After that, the link load correlation is considered by proposing an iterative method.

In [40], two types of grooming networks are distinguished, the constrained grooming networks where each node of the network has wavelength switching and subwavelength traffic grooming capability, and the sparse grooming networks where some nodes have limited grooming capability while others have none. The authors start with a simple two-hop single-wavelength capacity-correlated system with multi-rate traffic requests. Using these, a more complex and realistic multi-hop model is solved. The application of this model in performance analysis of general networks is also demonstrated. Because the capacity correlation model is specified for single wavelength systems, routing and wavelength assignment is not addressed in the survey.

Uncorrelated models

In some studies, a simplifying assumption of uncorrelated link loads is made. Xin et al. study the blocking performance analysis problem on traffic grooming in single hop mesh networks using such an uncorrelated model [35]. A closed-form formula is derived by some simplifications. For example, a single-wavelength link (SWL) blocking model is introduced and the multi-rate arrivals are converted into bulk arrivals and approximated departures. The authors also assume that overflow traffic is Poisson. Then a reduced load model is used to compute the end-to-end blocking probability.

By the same authors, the work in [36] is an extension of [35] that takes multi-hop routing into consideration. The authors propose a simple admission algorithm at a source node for each incoming traffic demand. A routing strategy is given such that the SWL model introduced in [35] can be extended to include multi-hop traffic arrivals. Instead of the sequential overflow model, a random selection of two-hop paths for the overflow multi-hop traffic demand is performed.

Yao et al. [39] also study the multi-hop case. Their model is as follows. For a given source-destination pair, some link-disjoint alternate paths are pre-determined and the s-d pair is blocked if all alternate paths are unable to carry it. On an alternate path, traffic can be electronically processed at some grooming nodes. To select grooming nodes, the authors introduce the load sharing policy, which tries the route without intermediate grooming nodes first and randomly select a candidate route if the direct one fails. Accordingly a path is blocked if all candidate routes are unable to satisfy it and respectively, a route is blocked if any of the lightpaths it consists are unable to satisfy it. Assuming the wavelength conversion capability is absent, a lightpath can be carried if there is an available single wavelength path (i.e., an available wavelength on all the links along the lightpath). The availability of a single wavelength path is in turn decided by the availability of the set of single wavelength links it consists, i. e., the existence of a set of common channels (time slots) that can satisfy the amount of capacity the s-d pair requires. Some restrictive assumptions are made in this study. First, the single wavelength links that make up a single wavelength path are assumed to be uncorrelated. Similarly, the lightpaths in a route are assumed to be uncorrelated. Finally, the overflow traffic is assumed to be Poisson.

Other models

The study in [43] deals with the traffic models in traffic grooming networks. The aim of the paper is to investigate how traffic elasticity, the reactivity of traffic with respect to the changing environment (load, e.g.), impacts grooming. The authors argue that even in core networks, the traffic is elastic in nature. Therefore, it is inappropriate to model them as the traditional circuit switched traffic. Specifically, two traffic grooming policies, one preferentially using existing lightpaths and the other preferentially setting up new lightpaths, are studied under two traffic models that have some feature of elastic traffic. The first model, which is less complex, captures the decrease of throughput of traffic when there is a congestion. The more complex model captures the fact that the more congested the network, the longer flows remain in the network. Different combinations of the traffic models and the grooming policies are simulated and compared. The simulation results show that the interaction between the IP and optical layer gives rise to some complex behaviors, which suggests that neither grooming policies are suited for the management of an IP over WDM grooming network, because they do not take the interaction between the IP and optical layer into consideration.

As we have mentioned in Section 2.2.5, [28] studies the "sliding scheduled traffic model". Specifically, a traffic demand is given by a tuple $\{s, t, n, l, r, \tau, p\}$, where s and t are the source and destination respectively, n is the bandwidth requirement, l and r are the starting and ending time respectively, and τ is the duration of the request; p is a binary parameter representing the priority of the demand. The traffic demand is required to be scheduled within the time window l to r (i.e., it should start between the time interval l to $r-\tau$); otherwise, it needs to be rearranged. Then the traffic grooming problem conceptually consists of two parts, the scheduling part and the grooming part. The scheduling part decides the starting time for each traffic demand in a manner such that the number of overlapping demand pairs in time is minimized. The grooming part then performs a time window based grooming algorithm. A space-time traffic grooming algorithm is proposed and compared with a tabu search algorithm that uses fixed alternate routing. The authors claim that the former algorithm outperforms the latter one in terms of the number of lightpaths.

2.3.3 Design Studies

Studies of the design problem usually contribute both in formulating the problem with a particular objective, and then obtaining a solution to it. Therefore, two major concerns are how accurate the formulation is and how amenable it is to a solution. In dynamic traffic grooming networks, one important challenge that impacts the accuracy of the model is how to model traffic variations. As we have seen in Table 2, different traffic models have been proposed. We have already mentioned that the problem can be formulated as an ILP when the traffic model is deterministic, or specifies that the traffic changes entirely to a new traffic matrix while the network is running [61].

Obtaining a solution corresponding to the formulation also poses a separate challenge. Since the general static traffic grooming problem is NP-Complete [66, 67], obviously the general dynamic traffic grooming problem with the static formulation, which has significantly more time dependant variables is also NP-Hard. It has been previously shown that the static problem may be even inapproximable [67]. Because of this, most research focuses on heuristic approaches.

Objective of minimizing blocking characteristics

As we have discussed in section 2.3.2, it is generally very hard (if not impossible) to find a closed-form solution to the blocking probability in grooming networks. Therefore, many researchers propose heursitic traffic grooming algorithms and compare their performance in terms of blocking probability. We distinguish different kinds of network characteristics desired in terms of blockings. In non-rearrangable networks, the algorithms can not disrupt the existing traffic. In rearangable networks, the existing traffic and/or virtual topology can be reconfigured or rerouted, which generally serves to lower the blocking.

4.3.1.1 Non-rearrangable Approaches

Because of the heuristic nature of grooming algorithms, some simple policies (or rules of thumb) may provide us some insight into the whole problem. When there is a new arrival, two simple and straightforward policies are (i) setting up a new lightpath or (ii) using the existing virtual topology. In [43], as we discussed in Section 2.3.2, these two policies are studied under two traffic models that have some feature of elastic traffic. The above grooming policies are classified as operation oriented policies in [51], which deals with the operations that will be performed to accommodate an arrival. Heuristic algorithms are advanced to decide between establishing a new lightpath or routing on the existing IP topology for an arriving LSP request. The decision is based on monitoring the current level of congestion in the network, because routing on the IP topology may increase congestion, but depending solely on new lightpaths will exhaust network resources quickly. In [53], the same authors extend the same idea to provide differentiated services based on the priority. High priority LSP requests should have lower blocking probability than the low priority requests. Low priority requests will be blocked if no lightpath can be set up due to the wavelength or transceiver limit. However, high priority requests will be routed on the IP/MPLS layer. The algorithm proposed is further modified by introducing the Average Path Inflation Index , which takes the holding time of LSP requests into account. The authors claim this algorithm can be extended to handle more than two priority classes.

Sabella et al. propose a strategy for dynamic routing in GMPLS networks [68]. The proposed strategy has two phases. First, an IP/MPLS topology is considered, where there is an MPLS link between two nodes if and only if there is at least one lightpath interconnecting them. Based on this topology, a proposed routing algorithm extends the least resistance routing weight method [48] to the multi-layer GMPLS paradigm, where subwavelength LSPs are routed. The second phase is the grooming phase, where the LSP is groomed into lightpaths. Two policies, the packing policy that prefers the most loaded lightpath and the spreading policy that prefers the least loaded lightpath, are addressed.

In [31], the authors use a genetic algorithm to find a grooming solution in a strictly non-blocking manner for all-to-all traffic demands. New traffic demands are satisfied without re-routing and reconfiguration. To realize the strictly non-blocking property, the chromosome is decoded by a first fit approach incorporated with a local greedy improvement algorithm.

4.3.1.2 Auxiliary Graph Algorithms

Several design studies to minimize blocking performance are distinguished by their use of a common methodology, and we discuss them separately in this section. These studies use an auxiliary graph of some sort constructed from the network graph, where information regarding other constraints or goals of the problem are embedded in the auxiliary graph. This approach takes the advantage of the flexibility of an auxiliary graph to use simple routing algorithms. The overall design algorithm can thus be expected to have low complexity and be suitable for on-line use, but can take cross-layer information and heterogeneous node architectures into consideration. Different studies propose different auxiliary graph constructions. Many of these studies are inspired by [49].

The study in [49] creates an auxiliary graph that has an access layer, a lightpath layer and W wavelengths layers, where W is the number of wavelengths on a fiber. Each layer has an input port and an output port. Different edges representing different node capabilities are inserted between ports. An edge has a property tuple that states its capacity and weight, which reflects the cost of each network element (transceiver, wavelength-link, wavelength converter, etc.), and/or a certain grooming policy. Different grooming policies are achieved by applying different weight-assignment functions to the auxiliary graph, which reflect various objectives. In [15], Zhu et al. study a more specific resource provisioning problem where network nodes have different grooming architectures. Without wavelength converters, the graph model proposed in [49] is simplified to consist of four layers, the access layer, the mux layer, the grooming layer and the wavelength layer. By splitting the lightpath layer in [49] into the mux and grooming layers, the model is able to support different types of lightpaths distinguished by the source and/or destination node grooming capabilities. Using this model, the authors illustrate how different traffic engineering optimization goals can be achieved through different grooming policies.

The auxiliary graph constructed in [56] has two layers, the virtual topology layer and the physical topology layer. An improvement to the previous work is the introduction of the link bundling (or more accurately wavelength bundling). In particular, following constraints are taken into consideration: the transceiver constraint and the generalized wavelength continuity constraint, which allows nodes equipped with different kinds of conversion capability. The Link Bundled Auxiliary Graph simplifies the auxiliary graph representation in [49] by aggregating at most the number of wavelengths available on a link into one arc. Based on the graph, an algorithm is proposed to find a feasible path and a feasible wavelength assignment. As multiple feasible paths may exist, grooming policies are introduced to select the preferred one.

In [50], Farahmand et al. propose the Drop-and-Continue node architecture, which, in addition to setting up some new lightpaths and/or utilizing existing lightpaths, allows two other operations, namely, drop-and-continue and lightpath extension. These two operations can reduce the network cost, especially for multi-cast networks where we have one source node and many receiving nodes. The auxiliary graph constructed has a dedicated layer for each wavelength and different edges describing existing lightpaths, potential lightpaths, potential extended lightpath and sub-lightpath. By assigning them different weights using different grooming policies that are essentially the same as those in [49], a shortest path algorithm is used to find the best solution. Note that without the intermediate dropping and extension capability, this algorithm becomes identical to that of [49].

In [20], the same authors use a similar approach, but specifically for unicast traffic. A difference of the constructions between this paper and [50] is the introduction of the grooming layer. Based on the auxiliary graph, a dynamic tree grooming algorithm, which has the weight assignment strategy (referred to as routing polices) as a sub-routine, is proposed. As in [50], the edges are assigned weights by different policies. Accordingly, the shortest path algorithm is used by the algorithm to setup a connection for an arrival. Specifically, the connection is set up either by establishing a new tree of drop-and-continue nodes along the vertices on the optical hop or extending an existing tree to cover the remaining vertices.

In [46], Two auxiliary graphs, namely the virtual graph and the layered graph, are introduced. In a virtual graph, the edges (so called partially available edges) represent the existing lightpaths that have spare capacity. The edges in a layered graph are fully available edges. Upon these two graphs, a two-layered routing algorithm is proposed, which tries to route a traffic demand on the virtual graph first, then tries the layered graph if the first step fails. Then, a single layered routing algorithm based on an integrated graph, which does not distinguish partially available edges and fully available edges, is proposed. The shortcoming of this algorithm is that using the shortest path algorithm on an integrated graph may result in a route that uses more new transceivers. A third algorithm is proposed to combine the advantages of the first two. The algorithms are compared in terms of the blocking probability and the joint routing algorithm outperforms the others irrespective of whether the number of transceivers is small or large.

In [59], Ho and Lee argue that, the algorithms proposed in [49] can be time-consuming in large scale mesh networks. A remedy is proposed by considering only part of the whole network when auxiliary graphs are constructed. Specifically, when a traffic demand arrives, instead of constructing an auxiliary graph with n nodes, where n is the number of nodes in the network, m candidates that are in the physical shortest path of the traffic demand are evaluated. If no lightpath can be found, neighbor nodes of the candidates are included into the consideration. This procedure can be repeated until a lightpath is found or resources are exhausted. In [58], based on the same idea, the authors propose a dynamic traffic grooming algorithm. In the first phase, to reduce the complexity of constructing an auxiliary graph of the entire network, a reachability graph that includes all the possible logical paths between the source and the destination is constructed. Based on the graph, the second phase is to find the optimal route by a cost-constraint algorithm, where the cost of interest is the sum of the cost of grooming fabrics and the penalty paid for wasted wavelength bandwidth.

4.3.1.3 Wide-Sense Non-blocking Studies

While all such studies aim at minimizing the blocking probability, in general the result is some low value of blocking. A few studies are able to propose approaches that result in a completely non-blocking network under specific conditions, and we discuss them separately here. The non-blocking achieved is *wide-sense*, that is the algorithm provided must be used exclusively for routing calls in that network, otherwise blocking may appear.

In [29], Sasaki and Gerstel study the dynamic traffic grooming problem for some typical WDM SONET ring architectures that guarantee no blocking. The primary network cost is the number of SADMs while the secondary concern is the number of wavelengths. For unidirectional path switched ring networks with limited number of wavelengths, a lower bound is derived by assuming the traffic is allowed to be cross-connected at every node. Given this lower bound, a single-hub architecture that guarantees wide-sense non-blocking, as well as a node grouped architecture designed for static traffic, are compared. For the wavelength limited case, the single-hub architecture and an incremental architecture are compared. The incremental architecture is a simplified version of the incremental network described in [60], where around the ring, nodes alternate between having the maximum and minimum number of ADMs. It shows that the incremental architecture is rearrangeably non-blocking and also wide-sense non-blocking for incremental traffic [60]. For two-fiber bidirectional line switched ring networks with unlimited number of wavelengths, the single-hub architecture is wide-sense non-blocking and it leads to a lowering of the bandwidth requirements because traffic may be routed on either direction of the ring. For the bidirectional wavelength limited case, the double-hub network is rearrangeably non-blocking [60], and the SADM cost is close to that of the single-hub network.

4.3.1.4 Rearrangable Approaches

The disruption caused by rearrangement of traffic being currently carried by the network may be an acceptable tradeoff for the reduced blocking that is usually obtained by using the added flexibility of rearrangement. In such studies, one main concern is when and how the network should be reconfigured to effect the rearrangement, and when rearragement should not be attempted.

Kandula and Sasaki study the dynamic traffic grooming problem with rearrangement on ring networks [44]. The authors provide a reconfiguration algorithm, called bridge-and-roll, such that the number of LTEs is reduced while keeping the network as bandwidth efficient as a fully opaque network. Putting different constraints on the resources, some interesting traffic models are introduced to illustrate the algorithm. In addition, to reduce the cost of traffic disruption, bounds are provided in terms of the number of reconfigurations.

In [62], Gencata and Mukherjee also study the reconfiguration problem. The traffic is assumed to fluctuate slowly compared to the observation period. In each observation period, the network load is monitored and compared with a high and a low watermark, which indicate if the link is under-utilized or congested. In an observation period, exactly one action, setting up a new lightpath or tearing down an lightpath, can take place. The duration of the observation period is adjustable to make a trade-off between efficiency and traffic disruption. If some links are congested, one new lightpath is setup in the observation period. If some links are under-utilized, one lightpath is torn down. The authors first formulate the problem as an MILP problem, with a goal to minimize the maximum load, with constraints that ensure the correct action is triggered and the virtual topology changes correspondingly. Then, the authors propose a heuristic adaptation algorithm. If some links are congested, the algorithm simply picks the link that has the maximum load and the maximum traffic component that traverses the link, then sets up a new lightpath for the selected traffic component. The authors use actual observed traffic traces for validation.

Based on a two-layer architecture, the authors of [18] propose an algorithm that allows rerouting existing traffic. It also allows segmented backup where multiple backup paths are allowed to share the same bandwidth. Traffic requests are multi-rate requests, and may or may not require protection. Therefore, to satisfy a new arrival with protection requirement, both the primary and backup routes need to be set up. In case rerouting existing traffic is necessary to accommodate a new arrival, end-to-end backup routes are considered first, in order to minimize disruption. If no route that is link-disjoint with the current primary and backup routes is found, all backup routes (end-to-end or segmented) are considered. The "best" route for a backup route is found if rerouting the backup on this route can satisfy the new arrival. Finally, existing traffic without protection requirements or with end-to-end backups are considered for dropping or rerouting.

The authors of [45] also study rerouting. When a traffic request arrives, rerouting is performed only if the existing routing fails to accommodate the request. The rerouting can be at lightpath level or connection level. The former changes the virtual topology while the latter only changes the routing of traffic on it. Both approaches have advantages and disadvantages. Lightpath rerouting may have lower time complexity because the input is the set of lightpaths, which are much fewer than the traffic requests. However, it is subject to a longer time of disruption because of the laser re-tuning time involved. Connection rerouting, although more complicated, provides a finer granularity of adjustment. Practically, a combination of both approaches may be more appropriate. Based on these two approaches at different layers, two algorithms are proposed. The first algorithm initially finds the set of critical wavelength of a path (the set of wavelengths that are used on only one link along the path), then the lightpath using this critical wavelength is rerouted such that a traffic request can traverse the path. Similarly, the latter algorithm finds the set of critical connections for a path and a connection request and reroutes the critical connection so that the new request can be satisfied.

Objective of maximizing fairness

Another objective of interest in traffic grooming networks is the fairness as we mentioned in Section 2.2.6. The main concern is that, traffic with lower bandwidth requirements should not starve traffic with higher bandwidth requirements, i.e., traffic with different bandwidth requirements should experience similar blocking performances. Otherwise a user transmitting a big file may have to choose to request a low bandwidth and take a longer time, in order to avoid blocking. Indeed, fairness is one of the important metrics of QOS, which is generally implemented by the Call Admission Control. While CAC comes under the general area of grooming policy design, it is a distinct area which has received significant attention and it is worth mentioning separately. As one of the major functionalities that the control plane needs to implement, CAC has been extensively studied in signaling-based networks (e.g., ATM), where a call is accepted or rejected with respect to a pre-established agreement between the user and the service provider or the resource availability. In the context of optical grooming networks, we expect that some “old” concepts (e.g., QOS) will be re-examined by taking the virtual layer into consideration. As we have mentioned, when a new call arrives, the basic actions to take are *accept* and *reject*. Without CAC, a call will be rejected only if the available resources are unable to accommodate it. However, in a network with service differentiations, this simple strategy may not lead to an optimal overall utilization/revenue.

In [30], a CAC algorithm is proposed to deal with the capacity fairness, which is achieved when the blocking probability of m calls of line-speed n is equal to the blocking probability of n calls of line-speed m, and this is true for every pair m, n of line-speed. The overall blocking probability is defined as the blocking probability per unit line-speed of the call requests. The fairness ratio F_r is defined as the ratio of the estimated blocking probabilities of calls of lowest and highest line-speeds. Therefore, the goal of the CAC algorithm is to make F_r as close to 1 as possible while keeping the overall blocking probability acceptable.

Mosharaf et al. study the CAC problem from the wavelength provisioning aspect in [33]. A simple 2-hop tandem network with three classes of wavelength requests, requests traversing the first hop only, requests traversing the second hop only and requests traversing both hops, is considered. This problem is formulated as a Markov Decision Process problem. When a wavelength request terminates, the network decides for which class this wavelength is reserved. The best policy (the set of best actions for each possible state) is achieved by the Policy Iteration algorithm which maximizes the overall weighted utilization, using the discount cost model with infinite horizon. In [41], the same authors extend the work of [33] to grooming networks where traffic demands are usually subwavelength, with the goal to minimize the unfairness. Considering a single-hop single wavelength network, traffic is classified according to the bandwidth it requires. Thus, the network state is described by the number of existing calls of each class. Using this simple model, the optimal policy is examined. The authors also propose a heuristic to decompose tandem and ring networks using the idea of pre-allocating wavelengths for traffic with different $o-d$ pairs such that overlapping $o-d$ pairs do not share wavelengths (note that this is possible because the routing for all $o-d$ pairs are predetermined in the ring and tandem topologies). The numerical results show that substantial improvement in terms of fairness and utilization can be achieved compared to that of complete sharing policy and complete partitioning policy.

In [52], the authors study the fairness problem based on an auxiliary graph model, which consists of wavelength planes and different kinds of edges, which represent the availability of wavelengths, availability of grooming capability, the availability of transceivers and the source and destination of the traffic demand. In addition to grooming policies, two fairness policies are proposed. The fairness is evaluated in terms of the blocking probabilities of traffic demands with heterogeneous requests. The first policy sets a wavelength quota for each class of connections, distinguished by the rates they request. Since traffic demands requesting higher rates are more likely to be blocked, these classes receive more quota. Based on the quota, a dynamic grooming algorithm called wavelength quota method is proposed. The next policy is transceiver quota policy. Instead of counting the wavelength quota, transceiver quota is used to groom heterogeneous traffic demands in a manner as fair as possible.

Objective of minimizing OEO

We noted in Section 2.2 that OEO minimization is a common theme of static grooming. In this section we discuss literature that addresses the same objective in the dynamic context.

In [26], Berry and Modiano address the dynamic traffic problems in SONET ring networks. The problem is defined as minimizing the number of ADMs while being able to satisfy a set of allowable traffic requests. The authors first lower bound the number of ADMs, which corresponds to the no grooming solution. A bipartite matching approach is then proposed to combine two solutions such that any one of the traffic requests can be satisfied while keeping the number of ADMs minimized. To study a specific and realistic dynamic traffic model, the t-allowable traffic model is introduced (see Section 2.2.5). The authors lower bound the number of ADMs and model it as a bipartite matching problem. Unnecessary ADMs are removed by applying a necessary and sufficient condition to support the t-allowable traffic. The authors extend the work to support dynamic traffic in a strictly non-blocking manner and show how hub nodes and tunability can further reduce the number of ADMs.

Hu studies the deterministic traffic model and present an ILP formulation with the goal to minimize the number of ADMs in [25]. Both unidirectional and bidirectional rings are studied. A nice observation for unidirectional rings proved in [25] is that the integer constraint for the variable x^r_{ijl}, the number of traffic circuits from node i to j in the rth traffic requirement that are multiplexed onto wavelength l, can be relaxed and turn the ILP into a MILP formulation that is easier to solve. Unfortunately, this is not true for the bidirectional case. Because of the routing problem involved (clockwise or counter-clockwise), the dynamic traffic grooming problem in bidirectional rings is much harder to solve. Some heuristic methods are proposed. Keeping the same set of constraints, the cost function is slightly modified by integrating the cost of wavelength and the cost of ADMs. The modified ILP formulation can provide an initial solution relatively easily. Then, a heuristic method is used to aggregate sub-wavelength circles into wavelengths.

To solve the same design problem, i.e, ring networks with a deterministic traffic model, in [57], two traffic splitting methods called traffic-cutting and traffic dividing are proposed to manipulate the traffic matrices. Starting from the all optical topology, the traffic-cutting method cuts the lightpath from source to destination at an intermediate node, combining the pieces with existing lightpaths as necessary to avoid requiring additional ADMs. The benefit is that, the traffic component can change its wavelength at the dropping node, which turns out to be more efficient in terms of the number of ADMs and wavelengths required. The traffic-dividing method allows traffic bifurcation, that is, different parts of a traffic component can be routed on different lightpaths. The authors propose a synthesized-splitting method that combines both the traffic-cutting method and the traffic-dividing method. A genetic algorithm is developed such that a given set of traffic matrices is satisfied in a strictly non-blocking manner.

In [27], the authors propose an ILP with the objective to minimize the number of transceivers for mesh networks, also under deterministic traffic, in the form of multiple traffic matrices. The ILP formulation explicitly rules out cycling of lightpath and routing. To solve the problem, a simple heuristic utilizing the time-varying state information is proposed. The sum of a traffic component's demands in every traffic matrix is used as a metric. Based on this metric, a traffic component is selected and either routed on the existing network or on a newly established lightpath (the choice is controlled by a predetermined parameter).

The study in [65] addresses the problem of deciding, based on the network state, when traffic grooming should be performed. The cost it considers is a function of the DXC ports and OXC ports. The network topology has two layers: the optical layer where optical express links (essentially, lightpaths) are connected by OXCs, and the physical layer where fiber links are connected by DXCs. Note that the OXCs and DXCs are physically decoupled. It is different from other studies where a grooming node is equipped with both an OXC and DXCs. A traffic request can be either routed on the physical topology (i.e., through DXCs) or on the logical topology (i.e., through OXCs). To decide between these alternatives, a parameter θ is defined as threshold, which should be tuned such that the overall cost is minimized. Both a centralized and a decentralized algorithms are proposed.

Kuri et al. study the mathematical model for *Scheduled Lightpath Demands* (SLDs) [19], which are in units of number of lightpaths. The cost is a function of the number of ports. By introducing the *Multi-Granularity Switching Optical Cross-Connects*, a waveband layer is inserted between the physical layer and the traffic demands. Hence in this context, grooming refers to aggregating and disaggregating lightpaths into waveband-switching connections of the virtual topology. Similar to the wavelength assignment problem in wavelength routed networks, SLDs are assigned routed scheduled band groups. Then, the SLD Routing (SR) problem and the SLD Routing and Grooming (SRG) problem are formulated as combinatorial optimization problems with the objective to minimize the cost. In [63], the authors extend the above work by taking subwavelength traffic demands into consideration. That is, a traffic demand can be decomposed into SLDs, that request a number of lightpaths, and a *Scheduled Electrical Demand* (SED), that requests part of a lightpath. The work is based on WDM networks with hybrid node architectures, i.e, a node consists of both an OXC and an EXC (same as a DXC). The problem is to find the size of the OXCs and EXCs that allow a network of a given topology to serve a given set of Scheduled Demands (SDs) at the lowest cost. To solve this problem for SEDs ,the authors propose a simulated annealing based routing and grooming strategy.

The study in [64] is an extension of a previous paper by the same authors, where dominating set algorithms are proposed to solve the problem of the placement of wavelength converters. Becuase nodes with grooming capability are expensive, in this paper, the authors show that by appropriately selecting nodes, benefits of full grooming can be achieved with comparatively few nodes actually equipped with grooming capability. The traffic model studied is non-uniform. This is done by randomly assigning different nodes different weights and nodes with higher weight values generate more traffic than others with less weight values. Thus, the problem is modeled as the sparse grooming problem and formulated by the K-weighted minimum dominating set of the graph. A distributed voting algorithm is proposed and messages that are exchanged among nodes are introduced. Using these messages, a *Master* is selected and serves as the grooming node.

Other approaches

In [38], the problem studied is a virtual topology design problem in mesh networks. The authors propose a formulation of the multi-hop dynamic traffic grooming problem, which aims at minimizing the network resource. The main difference of the formulation with those in other works is that the blocking probability is included as a constraint. The blocking model proposed is based on the concept of grooming links (g-links), where a g-link between two nodes is the set of possible lightpaths. A blocking model is proposed then to impose constraints on the number of lightpaths needed on g-links. The authors then present an ILP formulation that also imposes constraints on the maximum amount of by-pass traffic, the number of ports at each node, and the conversion capabilities.

In [54], the authors compare the performance and cost on different network architectures, the point-to-point network, single-hop network and multi-hop network. To take the network cost into account, the metric that is compared is in terms of the blocking probability versus the total arrival rate per dollar. To decide the cost for different network architectures, two steps are performed. First, the off-line network design step determines the hardware cost (the number of wavelengths, transmitters and receivers) for each architecture. After the off-line step, the on-line connection provisioning step that determines how the resources are used to accommodate dynamic traffic requests follows. In this step, a simple auxiliary graph based algorithm is used for each architecture. Simulation results show that multi-hop network is generally the best under a variety of cost scenarios. An interesting observation is that while the point-to-point architecture obviously has the lowest blocking probability, this is not the best choice if the cost of architecture is taken into account.

In [47], Elsayed addresses not only the dynamic routing and wavelength assignment problem but the fiber selection problem. The network studied has multiple fibers between each node pair. The original physical graph is folded out into W copies, where W is the number of wavelengths available on each fiber link. Since the nodes are wavelength-continuity constrained, these copies are isolated. Based on this layered graph, a modified Dijkstra's algorithm with reduced complexity is proposed to find a path for a source destination pair. Once the path is found, the fiber selection algorithm is called. Two selection methods, least-loaded fiber selection and best fitting fiber selection, are evaluated.

Srinivasan and Somani propose an extended Dijkstra's shortest path algorithm in WDM grooming networks [12]. Specifically, every node is assumed to be wavelength continuity constrained. The path vector is defined by the available capacity and hop-count. Two path vectors at a wavelength continuity constrained node are combined by taking the minimum capacity, which is different from the traditional Dijkstra's algorithm where costs are linear (i.e., additive). The authors then propose different policies to select paths based on the path vectors, namely Widest-Shortest Path Routing, Shortest-Widest Path Routing nd Available Shortest Path Routing. Finally, the algorithm is examined in terms of the request blocking probability, network utilization, average path length of an established connection, average shortest-path length of an accepted request and average capacity of an accepted request. In [55], the same authors make a comprehensive study on the comparative performance of different dynamic routing algorithms under different node architectures. The metrics in WDM grooming networks are classified as *concave* (e.g., the capacity of a path is the minimum capacity among the corresponding links), *additive* (e.g., the length of a path is the sum of the length of corresponding links), and *multiplicative* (e.g., the reliability of a path is the product of link reliabilities). Accordingly, depending on the node architecture, the link-state vectors are combined using different operations to form the path vectors. In addition to the approaches proposed in [12], a request-specific version of available shortest path routing is also studied. Finally, assuming traffic bifurcation is allowed, a dispersity routing algorithm is also evaluated. A counter-intuitive result noted is that increasing the grooming capability in network could degrade the performance under one of the algorithms.

The study in [21] addresses the algorithm design problem for multicast traffic in WDM grooming networks. The authors first introduce a node architecture that supports multicast traffic. To model the light-tree, a hypergraph logical topology is proposed, where a light-tree is represented as an arc (referred to as a hyperarc). For a multicast session, the destination nodes are represented by a supernode. Based on this hypergraph logical topology, the single-hop grooming approach and the multi-hop grooming approach are proposed. In the single-hop grooming, the hypergraph is searched for an available hyperarc for the new multicast request. In the multi-hop grooming, a hyperarc with the same supernode as the request and a single-hop lightpath from the source node of the request to the source node of the hyperarc are found. The multicast session is established on the combination of a single-hop lightpath and a light-tree.

Table 2.1: Variants of the Dynamic Grooming Problem

Analysis (of Blocking Probability)	Virtual topology is assumed to be ...	static, given	opaque [33, 34]
		dynamic, strategy given; call routing is...	single-hop [35]
			multi-hop [36, 33, 30] [17, 34, 37, 38, 39]
	Specific modeling technique ...	link loads assumed...	correlated [17, 34, 37] [30, 33, 40]
			uncorrelated [35, 36, 38, 39]
		traffic rate model	multi-rate Poisson [30] [35, 37, 34, 38, 36]
			single-rate Poisson [17, 33]
Design (Performance Optimization)	Traffic variation modeled as ...	arrival departure model	Poisson model [33, 41]
			incremental [42]
			elastic [43]
		traffic matrix constraints	peak constraint [26, 44]
	Objective of design is ...	blocking probability; network is...	non-rearrangable [26] [45, 46, 47, 48, 49, 50] [51, 52, 53, 54, 55, 56] [31, 57, 58, 59]
			wide sense [29, 60]
			rearrangeable [29, 60] [44, 61, 62, 45]
		fairness [52, 30, 12, 33, 41]	
		OEO costs, metric is...	number of LTEs [42] [57, 63, 64, 65, 29, 60] [26, 66, 25, 54, 27]
			number of wavelengths [29, 60, 25, 27, 38]
			amt. of OEO processing [67]
	Virtual topology in solution is allowed to be ...	static [68]	
		one per traffic pattern	
		sequence, schedule of virtual topologies [44]	

Table 2.2: Literature Summarization

[43]	Traffic Modeling. Elasticity of IP traffic impacts grooming algorithms.
[49]	Using an auxiliary graph with variable edge weights and groomig policies to achieve multiple goals.
[30]	Call Admission Control algorithm dealing with capacity fairness.
[51]	Various grooming policies combined with path inflation control.
[53]	Extend path inflation control to provide differientiated services.
[28]	A sliding window traffic model that introduces the scheduling problem to dynamic traffic grooming.
[21]	Traffic grooming for multicast traffic.
[59]	An auxiliary graph approach with low time complexity.
[58]	A two-phase dynamic grooming algorithm using simplified auxiliary graphs.
[55]	A comprehensive study of routing algorithms in traffic grooming networks.
[29]	Comparison of typical ring architectures that guarantee non-blocking.
[26]	Minimization of the number of ADMs for t-allowable traffic.
[57]	Heuristics to solve the design problem in rings with given traffic matrices.
[31]	A genetic algorithm for strictly non-blocking grooming in unidirectional rings.
[27]	Design in mesh network using traffic time-varying state information.
[38]	An ILP formulation for mesh networks taking the blocking probability into consideration.
[19]	Mathematical models for routing and grooming scheduled lightpath demands.
[63]	Extension of [19] by taking subwavelength traffic (scheduled electrical demand) into consideration.
[64]	The placement of grooming nodes formulated as a dominating set problem.
[56]	An auxiliary graph approach with link-bundling.
[45]	Rerouting algorithms for varying traffic demands.
[50]	Traffic grooming with a drop-and-continue node architecture.
[20]	An auxiliary graph approach that supports dynamic unicast traffic.
[46]	An auxiliary graph approach that introduces the virtual graph and layered graph.
[47]	Routing, wavelength assignment and fiber selection algorithms in multifiber WDM networks.
[44]	Dynamic traffic grooming with reconfiguration algorithms.
[65]	Making grooming decision by monitoring the link performance.
[62]	Reconfigurition of virtual-topology by monitoring the link load.
[68]	A layered dynamic routing strategy in GMPLS networks.
[33]	A Markov Decision Process model for dynamic wavelength allocation in 2-hop tandem networks.
[41]	Call Admission Control for subwavelength traffic demands formulated as Markov Decision Process Problems.
[52]	Auxiliary graph with fairness grooming policies.
[35]	Blocking probability in single-hop traffic grooming mesh networks.
[36]	Blocking probability in multi-hop traffic grooming mesh networks.
[34]	Blocking probability in tandem networks, exact solutions with multi-rate arrivals.

Chapter 3

Link Capacity Allocation

Resource provisioning problem has been studied extensively since the QoS (Quality of Service) provisioning became an important issue in networks. In most studies, it is assumed end-users' responsibility to keep track of the traffic dynamic and provide the network a description of the traffic characteristics, such as the peak rate, average rate, delay tolerance, etc. According to the description, the network will allocate the resource (bandwidth, buffer, etc.) or reject the request if the available resource is unable to satisfy the requirement. It turns out that when a user requires more resource after an allocation is done by the network, it has to request a resource re-provisioning. This re-provisioning is not guaranteed to be successful and the user may experience undesirable service disruptions. This seems to be a natural approach in many situations, especialy when the carried traffic is highly dynamic. If the user underestimated the resource requirement, it is the user's responsibility. From the network point of view, it will take little effort to re-allocate the resource. However, we show that in some optical networks where traffic is less dynamic, the reactive provisioning approach may be inefficient because of the re-configuration cost of optical devices. Therefore, to avoid frequent time-consuming adjustment of optical devices, the network shares the responsibility with users to decide how many resources should be allocated and when a re-configuration should be taken before the resources are exhausted.

Wavelength-routed optical networks have long been recognized as the backbone networks of tomorrow. The attractive qualities of optical transmission, together with the technology of routing individual wavelength channels without the need for intermediate

Opto-Electro-Optic (OEO) interconversion, has further made such technology attractive for a future in which the backbone will be characterized by the need for high speeds with highly predictable performance. The literature on the *virtual topology design* problem has focused on the possibility of forming *lightpaths* or clear optical channels that can then be viewed as traffic that must be routed and assigned wavelengths on the physical topology of the network [**?**].

In the course of evolution of next generation networks and networking applications, it has become clear that realistically end-to-end traffic demands are typically considerably smaller than the capacity of whole wavelength channels, and such *sub-wavelength* traffic must be multiplexed (using electronic TDM methods) into individual wavelength channels to obtain good utilization of network bandwidth. This process has been called *traffic grooming*, and the literature has seen a significant amount of interest in this area in recent years [**?**].

Traffic grooming is an example of the cross-layer design methodologies that have been more and more popular in recent literature, as the limitations and inefficiencies of layer-isolated design have become more well understood. This technique operates jointly on the mapping of the optical transport layer to the physical layer (lightpaths to fibers) and the mapping of the end-to-end traffic demands to the optical transport. In the terminology of GMPLS, a layer-isolated design may be considered as designing an *overlay* control plane, where the control planes of the optical and traffic (IP or similar) layers operate separately and without knowledge of each other. The other extreme alternative is *integrated* design, where the two control planes are integrated into one, a cross-layer approach. Traffic grooming falls into the second class.

Network design with sub-wavelength traffic demands has grown more and more worthy of consideration because of several reasons. Optical networks are being extended closer and closer to end users, where more flexibilities to set up and tear down low speed traffic commands are required. In the framework of GMPLS, for instance, signaling and routing protocols used in MPLS networks have already been extended to other networks including time-division, spatial and wavelength switching networks. LSPs (label switched paths) in lower hierarchy are allowed to be tunnelled through LSPs in higher hierarchy. In such a scenario, when traffic demands (setup signaling messages) reach the edge of the optical network, the GMPLS-aware *Optical Cross-Connect* (OXC) has to assign them lightpaths and route them by some kind of traffic grooming algorithm. A recent trend in traffic grooming literature has been to consider the emerging area of *dynamic traffic grooming*, in which traffic demands are not considered to form an absolutely static traffic matrix, but may change over time. Different models of variation have been considered, from a completely dynamic call arrival and departure model, to nearly static traffic matrices with a small amount of allowed variations in the magnitude of the traffic components. In [**?**], we have attempted to describe the different categories of assumptions regarding variation of traffic with time and allowed design choices. In this work, we consider a case where aggregrate traffic changes in magnitude over time, but in small increments and decrements rather than arrivals and departures of whole components. Such a traffic model has been called *quasi-static* to distinguish it from the arrival-departure model, which is traditionally called *dynamic*. Grooming time-varying traffic is becoming an increasingly important research area, because increasing network sophistication is likely to make networks lower in the aggregation hierarchy, where aggregation does not smooth out most variations in traffic, candidates for grooming strategies. Next-generation SONET (NG-SONET) networks are expected to carry time-varying traffic. To overcome the intrinsic inefficiency of carrying dynamic traffic demands, the mechanisms of Virtual Concatenation (VCAT) (as defined in [ITU-T G.707]) and Link Capacity Adjustment Scheme (LCAS) (as defined in [ITU-T G.7042]) have been developed. Using VCAT, non-contiguous Synchronous Payload Envelopes (SPEs) that form a Virtual Concatenation Group (VCG) can be routed on the physical topology independently and recombined at the destination. Complementary to VCAT, LCAS dynamically adjusts the capacity allocated to a traffic component by adding or removing members of its VCG.

3.1 Prior Work

As we have shown in the previous chapter, most of the studies on the dynamic traffic grooming problem consider discrete call arrival and departure models, and focus on the blocking probability experienced by sub-wavelength calls as the relevant performance metric.

In [44], the authors study the dynamic traffic grooming problem with rearrangement on ring networks. The dynamic traffic grooming problem in Mesh networks using a graph model is studied in [49]. The study in [46] also proposes some graph-based dynamic traffic grooming algorithms. The main constraints considered are the number of transceivers and wavelength continuity. The authors of [69] study the traffic grooming problem for WDM rings, both unidirectional and bidirectional, with the objective of minimizing the number of ADMs. A heuristic method in combination with the ILP formulation is proposed. The heuristic involves two phases, first, an ILP is solved to minimize the number of wavelengths; then, based on the solution provided by the first stage, "subwavelength circles" are combined into wavelengths such that the number of ADMs is reduced as much as possible. In [35, 36, 12, 34], the performance analysis problem related to dynamic traffic grooming is studied. Queueing models are used to determine the blocking probability under a given call arrival/departure model and network policy. We do not discuss them further here because our focus in the dissertation is a design problem, rather than one of analysis. In [**?**], we had proposed the over-provisioning method as an integrated approach to supporting dynamic traffic grooming, by balancing the reconfiguration cost and traffic grooming gain. This also improves network agility, or response time to change in traffic demands. However, the optimality of overprovisioning was not considered in [**?**]; instead naive rule-of-thumb approaches were used such as all-equal, proportional etc.

In this chapter, we focus on a specific spare capacity allocation in support of dynamic grooming. As we show in the next section, our problem can be viewed as a version of the packing problem. In the literature, both the integer version and the continuous versions of this problem have been extensively studied. In [**?**], Hauchbaum and Shanthikumar present an algorithm for the continuous version, which is polynomial in terms of the input size, required precision and the largest subdeterminant. They also show that for the integer version, the algorithm is polynomial provided that the corresponding integer linear version is polynomial. Therefore, the single link problem, is a special case that can be solved in polynomial time because the constraint possesses the total unimodularity property, that is, the subdeterminant is either $1, -1$ or 0. In [18], instead of using the widely used scale-and-iterate approach, Magnanti and Stratila present a polynomial approximation algorithm using a *single* piecewise linear approximation for each component of the objective function. It is shown that using pieces exponentially increasing in size, the polynomial solvability for separable concave problems is still guaranteed.

3.2 Our Contribution

In this chapter, we study a problem motivated in [?]. Given a current grooming solution in a network in which the traffic components can request increments and decrements at intervals, we address the question of how to pre-allocate any remaining spare capacity to allow the network to continue operating as long as possible before reconfiguring the logical topology in order to satisfy the requests. We show that this problem can be easily solved optimally in the single link case, but it is not straightforward to show this. We also show that the problem becomes difficult for general topologies, and provide good heuristics to perform the allocation in that case.

The rest of this chapter is organized as follows. The next Section formulates the problem precisely, and in the Sections 3.4 and 3.5 we address the single link and general topology cases. Section 3.6 presents our numerical results.

3.3 Problem Formulation

We consider an optical network with a logical topolgy currently implemented, and carrying traffic specified by a traffic demand matrix. The logical topology, the current traffic matrix and the current routing of each component of the traffic matrix on the logical links comprising the logical topology (that is, the complete current grooming solution) are given as part of the problem. Each arc of the logical topology graph is a continuous optical channel, traversing more than one physical fiber link in general, formed by wavelength switching at intermediate nodes. Such end-to-end optical channels have been called “lightpaths” in literature; in this work we refer to them as *logical links*, or simply *links* when the context makes it clear that logical (and not physical) links are under consideration. Each traffic demand component is specified as an integer, indicating multiples of some basic traffic rate (such as OC-3). In the same units, the capacity of each wavelength channel is denoted by C. Each traffic component can undergo some change in values, that is traffic components can request increment and decrement of the current magnitude. An increment request must be satisfied by allocating the increment *with the same routing over the logical topology*, to guarantee the same delay and jitter characteristics to all parts of the traffic flow. In our model, we allow each traffic component to request an increment or decrement of only one unit at a time, but the requests are assumed to come randomly. A decrement request can be satisfied at any time, but an increment request can be satisfied only if the traffic component has currently been overprovisioned by at least one unit. That is, we only count an *agile* admission of an increment request as successful. It is possible that an increment request can be put on hold while the logical topology and grooming is rearranged in order to allow the request to be satisfied, but this involves much higher delays in satisfying the request and we do not count such a request as satisfied, in this chapter. Following [**?**], we assume that reconfigurations can be undertaken only *between* successive requests, and the problem at hand is to decide how to allocate the spare capacity in each link of the logical topology as overprovision to the traffic components traversing that link. Note that the physical topology (and lightpath RWA on the physical topology) can also be considered given, but are not required in our formulation, as a consequence of the above focus. Thus our work is applicable to a broad range of networking situations where traffic flows are routed over a network and spare capacity can be allocated by overprovisioning.

We denote the amount of current traffic demands by the traffic matrix $T = [t_{sd}]$, where t_{sd} is the traffic demand from source node s to destination node d. For any individual traffic component, the increment and decrement of the traffic component are stochastic events, hence the time until an increment request occurs that can not be met is a random variable. We call this variable the *Time To Rejection* (TTR) of that traffic component. Clearly, the larger the spare capacity assigned to that traffic component, the larger the expected or Mean TTR (MTTR). Now considering the whole network, from the provider's point of view, any rejection of an increment request is undesirable. Thus we define the TTR of the entire traffic matrix to be the time until the first rejection of an increment request occurs for *any* of the traffic components. The goal of spare capacity assignment is to maximize the MTTR for the entire network, *i.e.* the expected time until the first such rejection occurs. Again, it is obvious that the MTTR depends on the spare capacity assignment.

Let $\mathcal{L}_i$ be the set of logical links traversed by traffic component i, and S_l be the spare capacity available on link l. We denote the spare capacity assigned to traffic component i on link l by $s_i^{(l)}$, and the MTTR of the traffic matrix under this spare capacity assignment by $Q(\{s_i^{(l)}\})$. Then, the problem can be formulated as the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & Q(\{s_i^{(l)}\}) & (3.1)\\ \text{subject to} \quad & \sum_{i:l\in\mathcal{L}_i} s_i^{(l)} \leq S_l, \forall l & (3.2)\\ & s_i^{(l)} \in Z_0^+, \forall i, l \in \mathcal{L}_i. & (3.3) \end{aligned}$$

This problem can be viewed as a special case of the packing problem. It is in general a difficult problem, and we show that this is true of our particular problem in Section 3.5. However, in Section 3.4, we show that it can be easily solved optimally for a single link, under reasonable assumptions. First, we adopt a traffic variation model and study the effect it has on the MTTR of a traffic component in isolation.

3.3.1 Memoryless Variation Model and MTTR

Naturally, the nature of the function $Q(\{s_i^{(l)}\})$ depends on the process governing increment and decrement request arrivals. In general, $Q(\{s_i^{(l)}\})$ is a non-linear function. In the absence of any basis for assuming that increment and decrement requests from the different traffic flows are correlated, we make the natural assumption that the requests from each traffic flow is an independent process. For each traffic component, we assume that the arrival process of increment requests is memoryless with rate γ, and that of decrement requests is similarly memoryless with rate μ. To make the problem general, suppose that there is a lower bound (LB_{sd}) and an upper bound (UB_{sd}) on the amount of traffic component $t_{(sd)}$, that is, t_{sd} must be within the closed interval $[LB_{sd}, UB_{sd}]$, such that $\infty \geq UB_{sd} \geq LB_{sd} \geq 0$. Then a simple Markov model suffices to represent the behavior of any traffic component. Each state t in the closed Markov chain represents the case when the current value of the traffic component is t. Since the chain is defined only on values of $t_{(sd)}$ within the allowed limits, the lower bound serves only to change the labels of the states; and henceforth we only consider $LB_{sd} = 0$ without any loss of generality. Define the matrix $\mathcal{M} = [\mathcal{M}_{i,j}]$, where the $[i, j]$-th element $\mathcal{M}_{i,j}$ is the *first passage time* from state i to state $j + 1$. Note that the first passage time from a given state is *only defined to higher states*, thus the elements of the matrix below the diagonal do not exist or are not defined. Let s units spare capacity be allocated to this traffic demand (that is, a total of $t+s$ units of capacity is allocated for this demand). It is clear that an increment request can be satisfied as the traffic demand evolves, through increment and decrement requests, as long as the state is below $t+s$. However, an increment request when the current state is $t+s$ cannot be satisfied and will cause the first rejection. Note that the time to rejection becomes infinity if the amount of spare capacity is assigned such that $t+s \geq UB_{sd}$. For notational convenience, we denote $\mathcal{M}_{i,j}$ by $m(i, j-i)$. This allows us to conveniently represent the mean time until a traffic component in state t and allocated s units of overprovision grows beyond its provisioning as $m(t,s)$. It is now obvious that the quantity $m(t,s)$ represents the MTTR of the traffic component. Fig. 3.1 shows the Markov model for the variation of

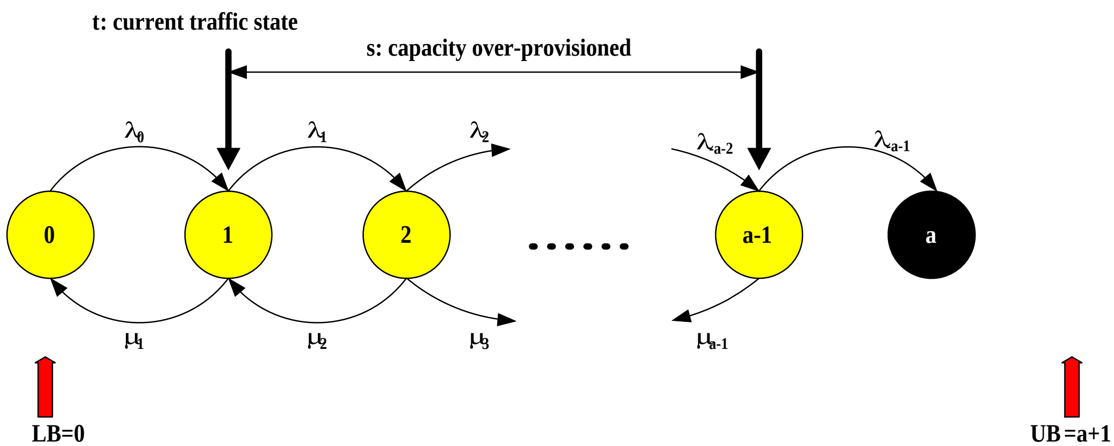


Figure 3.1: A Markov Chain model

this traffic demand.

From the Markov model, it is obvious that $m(0,0) = \frac{1}{\gamma}$. We can then find the further elements of $\mathcal{M}$ by the following recursive arguments. First, we observe that:

$$m(t,s) = m(t,s-1) + m(t+s,0) \quad \forall t \geq 0, s > 0, \tag{3.4}$$

which simply states that the mean first passage time from state t to $t+s+1$ is equal to the mean first passage time from state t to $t+s$ plus the mean first passage time from state $t+s$ to $t+s+1$. The mean first passage time from state t to $t+1$ can be derived as follows:

$$m(t,0) = \frac{1}{\mu+\gamma} + \frac{\mu}{\mu+\gamma} m(t-1,1), \tag{3.5}$$

where the first term is the mean sojourn time at state t, the second term is the probability that there is a departure times the mean first passage time from state $t-1$ to $t+1$. Using

(3.4) we obtain

$$m(t-1,1) = m(t-1,0) + m(t,0). \tag{3.6}$$

Finally, substituting (3.6) into (3.5),

$$m(t,0) = \frac{1}{\gamma} + \frac{\mu}{\gamma} m(t-1,0) \quad \forall t > 0. \tag{3.7}$$

Thus we obtain the following structure for $\mathcal{M}$:

$$\begin{pmatrix} \frac{1}{\gamma} & \frac{2}{\gamma}+\frac{\mu}{\gamma^2} & \frac{3}{\gamma}+\frac{2\mu}{\gamma^2}+\frac{\mu^2}{\gamma^3} & \frac{4}{\gamma}+\frac{3\mu}{\gamma^2}+\frac{2\mu^2}{\gamma^3}+\frac{\mu^3}{\gamma^4} & \cdots \\ - & \frac{1}{\gamma}+\frac{\mu}{\gamma^2} & \frac{2}{\gamma}+\frac{2\mu}{\gamma^2}+\frac{\mu^2}{\gamma^3} & \frac{3}{\gamma}+\frac{3\mu}{\gamma^2}+\frac{2\mu^2}{\gamma^3}+\frac{\mu^3}{\gamma^4} & \cdots \\ - & - & \frac{1}{\gamma}+\frac{\mu}{\gamma^2}+\frac{\mu^2}{\gamma^3} & \frac{2}{\gamma}+\frac{2\mu}{\gamma^2}+\frac{2\mu^2}{\gamma^3}+\frac{\mu^3}{\gamma^4} & \cdots \\ - & - & - & \frac{1}{\gamma}+\frac{\mu}{\gamma^2}+\frac{\mu^2}{\gamma^3}+\frac{\mu^3}{\gamma^4} & \cdots \\ & & \cdots & & \end{pmatrix}$$

Specifically, we have

$$m(t,s) = \begin{array}{l} \frac{s+1}{\gamma} + \frac{(s+1)\mu}{\gamma^2} + \cdots \\ + \frac{(s+1)\mu^t}{\gamma^{t+1}} + \frac{(s+2)\mu^{t+1}}{\gamma^{t+2}} + \cdots + \frac{\mu^{t+s}}{\gamma^{t+s+1}} \end{array} \tag{3.8}$$

$$= \begin{array}{l} \frac{s+1}{\gamma} \sum_{i=1}^{t} \left(\frac{\mu}{\gamma}\right)^i \\ \\ + \frac{\left(\frac{\mu}{\gamma}\right)^t}{\mu} \left((s+2) \sum_{i=1}^{s+1} \left(\frac{\mu}{\gamma}\right)^i - \sum_{i=1}^{s+1} \left(\frac{\mu}{\gamma}\right)^i i \right) \end{array} \tag{3.9}$$

Fig. 3.2 shows an example of $m(t,s)$ as a function of s, generated with $\gamma = .5$ per minute, $\mu = .55$ per minute and at $t = 0, 5, 10$.

While the above allows us to find the mean of the TTR, we also need its distribution to proceed. The distribution of first passage time is a complex and well-investigated topic. For our purposes, we are interested in the practical result that under many circumstances, the first passage time to a set of reasonably rare states is well approximated by an exponential distribution. This result was first shown in [70], and has been used in many different contexts since. The result follows from another and more general one: a geometric summation of IID RVs converges in distribution to the exponential as the termination probability of the geometric tends to zero from above. The method requires that the mean of the distribution be found independently of the exponential approximation assumption, as we have done above. The authority in [70] also recommends that the closeness of approximation

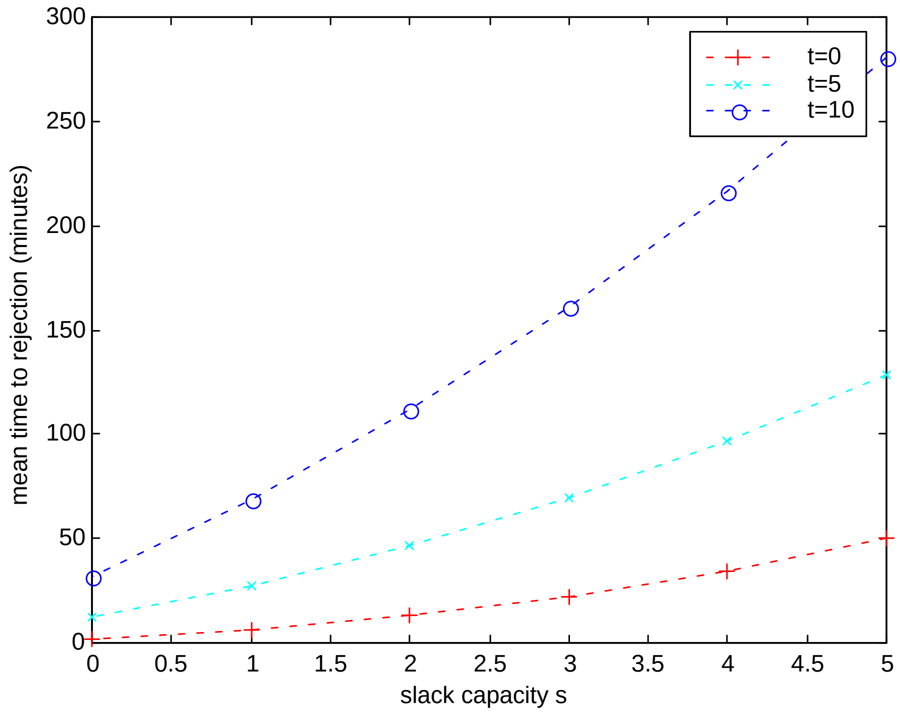
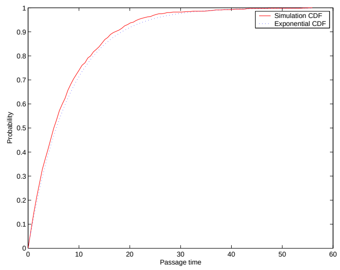


Figure 3.2: MTTR as a function of spare capacity

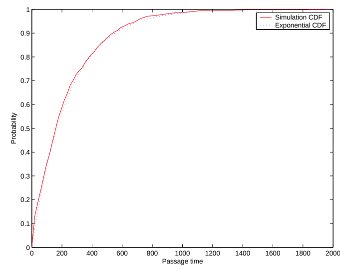
be verified by numerical experimentation. In Fig. 3.3(a)-3.3(c) we show a representative sample of such experimentation for our case, which shows that the approximation holds well for our problem; accordingly, we proceed with this approximation.

3.4 The Single Logical Link Case

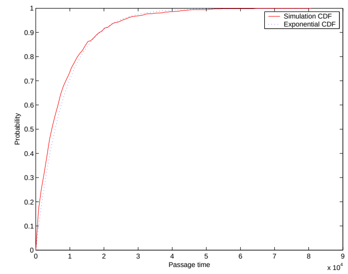
In this section, we study the case of a single logical link traversed by several traffic components. We start by making observations on the nature of the MTTR which enable us to subsequently design an algorithm for spare capacity allocation. Initially we assume $UB_{sd} = \infty$, later we remark briefly on how to take finite upper bounds into account.



(a) $t = 0, s = 1, \lambda = 0.5, \mu = 1$



(b) $t = 0, s = 5, \lambda = 0.5, \mu = 1$



(c) $t = 0, s = 10, \lambda = 0.5, \mu = 1$

Figure 3.3: Closeness of the exponential approximation for the TTR

3.4.1 A Convex Objective Function

In order to investigate the properties of MTTR, it is convenient to extend s and t to the real domain. We denote the continuous analogue of the discrete variable s as σ, of t as τ, and let $m(\tau, \sigma)$ denote the MTTR for a traffic component of magnitude τ to which σ spare capacity is allocated. We also denote $r = \mu/\gamma$. *In this section only*, since only one link is being considered, *we drop the superscript indicating the link* from the notation for the spare capacity assignment, that is we speak of s_i instead of $s_i^{(l)}$, similarly we speak only of σ_i.

If $r \neq 1$, by substitution in (3.8) and using standard geometric summation identities, we obtain

$$m(\tau, \sigma) = \frac{1}{\gamma(r-1)^2}\left[(\sigma+1)(1-r) + r^{\tau+1}(r^{\sigma+1}-1)\right]. \tag{3.10}$$

If $r = 1$, it is easy to show, again by substitution in (3.8),

$$m(\tau, \sigma) = \frac{(\sigma+1)(\sigma+2\tau+2)}{2\gamma}. \tag{3.11}$$

We consider a link l, and denote the set of traffic components traversing this link by $\mathcal{T}$ and $n = |\mathcal{T}|$. Let τ_i be the current amount of traffic demand $i, i \in \{1 \dots n\}$. Let the total amount of spare capacity on the link be S_l, that is $S_l = C - \sum \tau_i$. The problem is to determine the spare capacity σ_i to be assigned to each traffic component so that the MTTR for the whole link is maximized. Because spare capacity is assigned individually to each traffic component and not shared, the TTRs for the individual traffic components are independent random variables. From the last section, we know that each TTR can be considered to be exponentially distributed. By a well-known result, the minimum of independent exponentially distributed random variables is also an exponentially distributed random variable, with rate equal to the sum of the individual rates [?]. Thus the MTTR of the set of traffic components traversing the link l is given by the harmonic sum of the individual MTTRs:

$$Q_l(\tilde{\sigma}) = \frac{1}{\sum_{i=1}^{n} \frac{1}{m_i(\tau_i, \sigma_i)}} = \frac{1}{z_l(\tilde{\sigma})}. \tag{3.12}$$

Thus the optimization problem, in terms of the continuous rather than the actual integral quantities, becomes:

$$\text{minimize} \quad z_l(\tilde{\sigma}) = \sum_i \frac{1}{m_i(\tau_i, \sigma_i)} \tag{3.13}$$

$$\text{subject to} \quad \sum \sigma_i = S_l, \tag{3.14}$$

$$\sigma_i \geq 0 \forall i. \tag{3.15}$$

Note that we have expressed the problem as being one of minimization rather than maximization, and thus have been able to use the simpler function of the *rejection rate* $z_l(\cdot)$ rather than the MTTR $Q_l(\cdot)$. This is a non-linear objective function, but fortunately, it is an easy one in the practical sense, as the proposition below shows.

Proposition 1. *The objective function z_l as given by (3.12) is a strictly convex function of $\tilde{\sigma} = [\sigma_1, \cdots, \sigma_n]$.*

The proof of the above is tedious but straightforward, and is omitted here for the sake of brevity. It consists of showing that $1/m_i(\tau_i, \sigma_i)$ is convex, which is obvious for $r = 1$ and can be shown for $r \neq 1$ by taking the second partial derivative and decomposing it into parts until every part can be shown to be positive or zero. Then $z_l(\tilde{\sigma})$ is convex separable in the sense that it is the sum of convex functions $1/m_i(\tau_i, \sigma_i)$, each σ_i is an independent coordinate. The rejection rate of a single traffic component, obtained with $\gamma = 0.5$ per minute, $r = 2, \tau = 0$, is shown in Figure 3.4.

As we see in the following section, this characteristic of the problem allows us to prove the optimality of a quite simple approach. However, the convexity depends on the assumption we made regarding the state independence of γ and μ. We have investigated this issue in detail under other conditions, and can report the following: convexity is retained

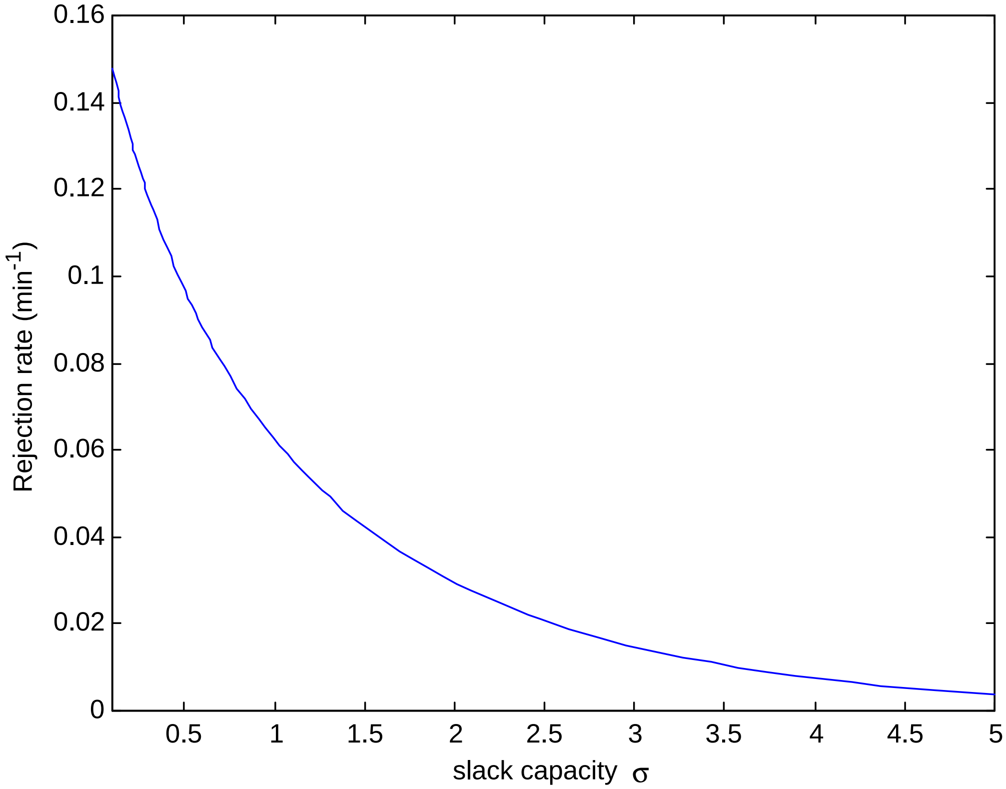


Figure 3.4: An example of function 1/MTTR, continuous version

even when the increment request rate and decrement request rates γ_t and μ_t are different for different states t, if $r_t = \mu_t/\gamma_t < 1$ for all states t, and $m_{t-1,t-1} \geq \frac{1}{\gamma_t - \mu_t}$. If either condition is violated in the general case of state-dependent γ_t and μ_t, then instances exist that show the objective is neither convex nor concave. Fortunately, our assumption of state-independence is quite reasonable, and indeed perhaps the most natural assumption in the absence of any specifc knowledge about the nature of traffic variation with time, so our conclusions continue to be valuable. We proceed with this assumption.

3.4.2 An Optimal Algorithm

The actual allocation of spare capacity to traffic components must obviously be made in the integer domain. The space of spare capacity allocation solutions contains

${}^{S_l}C_{S_l+n-1}$ possibilities, which is exponential. However, the convexity of the continuous form shows that the corresponding continuous problem is easy, and this encourages us to look for an optimal integer solution algorithm that is guaranteed to terminate in polynomial time. Indeed, we are able to show that a simple greedy approach is optimal, as a consequence of convexity. The key observation is that since the function is convex in the continuous domain, the convexity property carries over to the piecewise linear representation of the function defined on the discrete integer values. First, we show how an optimal algorithm may be constructed by recourse to the continuous version and rounding, but this results in a more time-consuming algorithm. Then we show how a much simpler algoithm can be also shown to be complex, using the insights obtained from the previous one.

In what follows, we relax the notation for more convenient representation, as follows. For a given problem instance, the current state of a traffic component t is an input, and remains fixed; whereas the spare capacity s to be assigned to it is what our algorithm needs to determine. In optimizational terms, t is a *parameter*, and s is the *variable* of interest. Accordingly, the effect of t on the MTTR of a traffic component is fixed as far as the algorithm is concerned. We are interested rather in the effect of variation in the assigned spare capacity s on the MTTR $m(t, s)$. In keeping with this point of view, in the simplified notation below, we use $m(s)$ in place of $m(t, s)$. Similarly we use $m(\sigma)$ in place of the more complete $m(\tau, \sigma)$, when discussing the corresponding continuous variables. Additionally, we use d_i to denote the i-th element of a vector $\tilde{d}$ of n elements. Finally, $\tilde{d}'$ denotes a vector such that $\sum_i d'_i = 0$, and $\tilde{d}_k$ denotes a vector of n elements for which the k-th element is 1 and all other elements are zero. Below we will use these as follows: $\tilde{d}'$ denotes a rearrangement of spare capacity among the n components, and $\tilde{d}_k$ denotes the addition of one new unit of spare capacity to some traffic component.

Since the objective function of the continuous version problem is convex as demonstrated above, the feasible region is also convex, and we can use standard nonlinear programming method to find the global optimal solution. Algorithm A1 below is based on utilizing this to obtain a solution to the integer version of the problem as follows:

Algorithm **A1**:

1. Solve the continuous version problem, let $\tilde{\sigma}^*$ be the optimal solution, if $\tilde{\sigma}^*$ is an integer solution, stop; otherwise let $s_i = \lfloor \sigma_i^* \rfloor, \forall i$

2. For $j = 1$ to $S_l - \sum_{i=1}^{n} \lfloor s_i \rfloor$,

3. $\tilde{s} \leftarrow \tilde{s} + \tilde{d}_k : k = \arg\min_l z(\tilde{s} + \tilde{d}_l), \forall l$

4. If $\exists s_i > \lceil \sigma_i^* \rceil$, stop; otherwise continue

5. Define $\mathcal{C} = \{i : s_i = \lfloor \sigma_i^* \rfloor\}$, if $\mathcal{C}$ is empty, stop; otherwise, let $m = \arg\min_l z(\tilde{s} - \tilde{d}_l) \forall l : s_l \leq \lfloor \sigma_i^* \rfloor$, $n = \arg\min_l z(\tilde{s} + \tilde{d}_l) \forall l \in \mathcal{C} \backslash m$

6. If $z(\tilde{s} - \tilde{d}_m) - z(\tilde{s}) > z(\tilde{s}) - z(\tilde{s} + \tilde{d}_n)$, stop; otherwise $\tilde{s} \leftarrow \tilde{s} - \tilde{d}_m + \tilde{d}_n$, go back step 5.

It is easy to see that step $2 - 6$ will terminate in at most n rounds. Next, we show that this algorithm will give us the optimal solution. The algorithm has three phases, the first phase, that is step 1, finds the optimal solution to the continuous version of the problem, then rounds them down to the nearest integers. After the first phase, we will have some unassigned spare capacity. In phase 2, that is step $2 - 4$, we assign these unassigned spare capacity to traffic components greedily. We will show in the following propositions that if a traffic component is assigned with more than one unit spare capacity, the algorithm stops after step 4 and gives us the optimal solution. Otherwise, the algorithm enters phase 3, that is, for all traffic components that were not assigned with spare capacity in phase 2 (i.e., for all i that $\lfloor \sigma_i^* \rfloor$ units spare capacity are assigned), we see if some perturbation of the assignment of these traffic components can make any improvement. This is done by sending as many i to $\lceil \sigma_i^* \rceil$ as possible. We first show a proposition that states, if $\tilde{s}$ is optimal, there may exist a traffic component i such that $s_i < \lfloor \sigma_i^* \rfloor$ or a traffic component j such that $s_j > \lceil \sigma_j^* \rceil$, but not both.

Proposition 2. *Integer solution $\tilde{s}$ is not optimal if $\exists i, j : s_i - \sigma_i^* < 1$ and $s_j - \sigma_j^* > 1$.*

Proof. Assume $\tilde{s}$ is the optimal integer solution and $\exists i, j : s_i - \sigma_i^* < 1, s_j - \sigma_j^* > 1$.

Without loss of generality, we assume $\sigma_j^* > 0$. Then, according to the KKT condition, we have

$$\frac{d}{ds_i}\left(\frac{1}{m_i(\sigma_i^*)}\right) = \frac{d}{ds_j}\left(\frac{1}{m_j(\sigma_j^*)}\right) \tag{3.16}$$

Because $1/m_i(\sigma_i)\forall i$ is a convex and decreasing function of σ_i, we have

$$\begin{aligned}\frac{1}{m_i(s_i)} - \frac{1}{m_i(s_i+1)} \geq \frac{d}{ds_i}\left(\frac{1}{m_i(\sigma_i^*)}\right) = \frac{d}{ds_j}\left(\frac{1}{m_j(\sigma_j^*)}\right) \\ \geq \frac{1}{m_j(s_j-1)} - \frac{1}{m_j(s_j)}.\end{aligned} \tag{3.17}$$

Thus, a contradiction to the assumption. ■

Then, we show step 4 will provide the optimal integer solution if it stops.

Proposition 3. *Let the solution produced by step 4 be $\tilde{s}$, then $z(\tilde{s}) \leq z(\tilde{s} + \tilde{d}')$, where $\tilde{d}'$ is a vector such that $\sum_i d'_i = 0$.*

Proof. Let k be any traffic component such that $s_k > \lceil \sigma_i^* \rceil$, we show that no $\tilde{d}'$ exists such that $\tilde{s} + \tilde{d}'$ is a better solution than $\tilde{s}$. We distinguish two cases.

Case 1. $s_i + d'_i \geq \lfloor \sigma_i^* \rfloor, \forall i$. This case can be put this way, after step 1, we get $\tilde{s}^{(0)} = [\lfloor \sigma_i^* \rfloor]$, then we have $S_l - \sum \lfloor \sigma_i^* \rfloor$ units of spare capacity to assign. In every iteration, our algorithm assigns one unit to a traffic component greedily until all $S_l - \sum \lfloor \sigma_i^* \rfloor$ units are assigned. The optimality of this greedy assignment actually motivates us the greedy algorithm in the next sub-session, where an induction proof showing the optimality of this greedy assignment is also presented.

Case 2. $\exists i : s_i + d'_i < \lfloor \sigma_i^* \rfloor$. Because of Proposition 2, we have $d'_k < 0$. Therefore, $\exists m : d'_m = 1$ and $s_m = \lfloor \sigma_m^* \rfloor$. However, because of the algorithm, we have

$$\frac{1}{s_m} - \frac{1}{s_m + 1} < \frac{1}{s_k - 1} - \frac{1}{s_k}.$$

This completes the proof. ■

Proposition 4. *The solution provided by algorithm **I** is an optimal solution to the integer version of the problem.*

Proof. We have shown the solution is optimal if the algorithm stops at step 4, now we show it is optimal if it stops at step 6. The algorithm will stop when either of the following two conditions is satisfied.

1. $z(\tilde{s} - \tilde{d}_m) - z(\tilde{s}) > z(\tilde{s}) - z(\tilde{s} + \tilde{d}_n)$, where

   $m = \arg\min_l z(\tilde{s} - \tilde{d}_l) \forall l : s_l \leq \lfloor \sigma_i^* \rfloor$, and

   $n = \arg\min_l z(\tilde{s} + \tilde{d}_l) \forall l \in \mathcal{C} \backslash m$

2. $\mathcal{C}$ is empty,

We discuss them separately.

1. If condition 1 is satisfied, it is easy to obtain that $z(\tilde{s}+\tilde{d}') \geq z(\tilde{s}) \forall \tilde{d}'$ because by sending more traffic components to their ceilings, we can not achieve any improvement.

2. If condition 2 is satisfied, then we have two sets $\mathcal{A} = \{i : s_i < \lfloor \sigma_i^* \rfloor\}$ and $\mathcal{B} = \{i : s_i = \lceil \sigma_i^* \rceil\}$. The following cases are discussed separately.

A. If $\tilde{d}'$ is such that $\exists i : i \in \mathcal{A}$ and $d'_i < 0$ and $\exists j : j \in \mathcal{B}$ and $d'_j > 0$, because of Proposition 2, $z(\tilde{s} + \tilde{d}')$ is not optimal.

B. If $\tilde{d}'$ is such that $\exists i : i \in \mathcal{A}$ and $d'_i > 0$ and $\exists j : j \in \mathcal{B}$ and $d'_j < 0$, $z(\tilde{s} + \tilde{d}')$ is not optimal. The reason is that we had sent s_j to its ceiling because it was more beneficial to take 1 unit spare capacity from i and assign it to j.

C. If the above conditions are not satisfied, we must have: $\exists i, j : i, j \in \mathcal{A}$ and $d'_i > 0, d'_j < 0$ or $\exists i, j : i, j \in \mathcal{B}$ and $d'_i > 0, d'_j < 0$. Because of condition 1, the later possibility is ruled out. For the first case, because that if we were taking 1 unit spare capacity from i and assigning it to someone else, we picked the i such that $i = \arg\min_l z(\tilde{s} - \tilde{d}_l) \forall l : s_l \leq \lfloor \sigma_i^* \rfloor$, that is, taking 1 unit spare capacity from j to i will not make any improvement of the objective. Again, we rule out the possibility that this case will lead us an optimal solution. ■

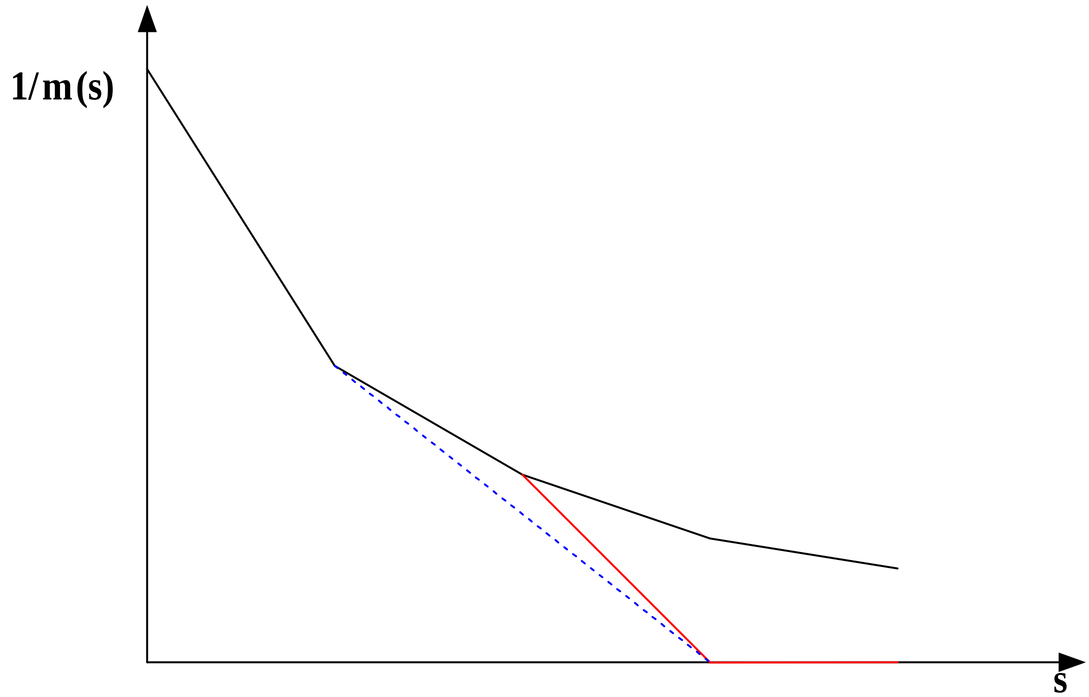


Figure 3.5: Loss of convexity with finite upper bound

Finally, we state how the case of finite upper bounds is addressed. Recall that we have assumed $UB = \infty$ in the convexity analysis. As shown in Fig. 3.5, a finite upper bound can change the MTTR to infinity if the upper bound of the traffic demand is less than the spare capacity assigned to the traffic component, thus may destroy the convexity. The reason for the complication of checking beyond the floors and ceilings of the continuous optimal allocations arises because these special cases allow us to sidestep the issue of this terminal non-convexity, because it would require *two* successive intervals to violate convexity for (3.17) to fail.

Now, we present the simpler algorithm which applies the same idea of adding one unit of slack successively, but does so greedily from scratch, without reference to the continuous version.

Algorithm **Alloc-1-link**:

1. Set $\tilde{s} : s_i = 0 \quad \forall i$

2. For $x = 1$ to S_l,

3. $\quad \tilde{s} \leftarrow \tilde{s} + \tilde{d}_k : k = \text{argmin}_l \, z_l(\tilde{s} + \tilde{d}_l), \forall l$

Obviously, this algorithm will terminate. Next, we show that $\tilde{s}$ gives us an optimal solution at the termination of the algorithm.

Proposition 5. *Let $\tilde{s}$ be the solution given by algorithm **Alloc-1-link**. Then, we have*

$$z_l(\tilde{s}) \leq z_l(\tilde{s} + \tilde{d}') \quad \forall \tilde{d}'. \tag{3.18}$$

Proof. Our proof is by induction; we show that after the c-th execution of Step 3, $\tilde{s}$ comprises an optimal allocation of c units of spare capacity to the n traffic components. At the first step, it is easy to see that algorithm **Alloc-1-link** will provide us the optimal solution, therefore (3.18) holds and the basis is obtained. At a step $c \in [2, S_l]$, we show that this optimality is maintained. The difference vector $\tilde{d}'$ can be interpreted as a feasible direction since $\sum_i (s_i + d_i') = \sum_i (s_i)$. Since the function $1/m_i(\sigma_i), i = 1, \cdots, n$ is convex and decreasing in σ_i, we have

$$\left. \begin{array}{l} \frac{1}{m_i(s_i - d_i)} - \frac{1}{m_i(s_i - d_i + 1)} \\ \geq \frac{1}{m_i(s_i - 1)} - \frac{1}{m_i(s_i)} \end{array} \right\} \quad \forall d_i > 0, \tag{3.19}$$

$$\left. \begin{array}{l} \frac{1}{m_k(s_k)} - \frac{1}{m_k(s_k + 1)} \\ \geq \frac{1}{m_k(s_k + d_k)} - \frac{1}{m_k(s_k + 1 + d_k)} \end{array} \right\} \quad \forall d_k > 0. \tag{3.20}$$

By the choice of k in Step 3,

$$\frac{1}{m_i(s_i - 1)} - \frac{1}{m_i(s_i)} \geq \frac{1}{m_k(s_k)} - \frac{1}{m_k(s_k + 1)} \, \forall i \neq k. \tag{3.21}$$

Now, we need to show that

$$z_l(\tilde{s} + \tilde{d}_k + \tilde{d}') \geq z_l(\tilde{s} + \tilde{d}_k) \forall \tilde{d}'. \tag{3.22}$$

We separate it into three cases:

1. $d'_k = 0$. From (3.18), we have

$$\sum_i \frac{1}{m_i(s_i + d'_i)} \geq \sum_i \frac{1}{m_i(s_i)} \tag{3.23}$$

$$\Rightarrow \quad \frac{1}{m_k(s_k)} + \sum_{i \neq k} \frac{1}{m_i(s_i+d'_i)} \geq \frac{1}{m_k(s_k)} + \sum_{i \neq k} \frac{1}{m_i(s_i)} \tag{3.24}$$

$$\Rightarrow \quad \frac{1}{m_k(s_k+1)} + \sum_{i \neq k} \frac{1}{m_i(s_i+d'_i)} \geq \frac{1}{m_k(s_k+1)} + \sum_{i \neq k} \frac{1}{m_i(s_i)}. \tag{3.25}$$

Therefore (3.22) is obtained.

2. $d'_k > 0$. Then, $\exists j : d_j < 0, j \neq k$, and using (3.20) and (3.21), we have

$$\frac{1}{m_j(s_j+d'_j-1)} - \frac{1}{m_j(s_j+d'_j)} \geq \frac{1}{m_k(s_k+1+d'_k-1)} - \frac{1}{m_k(s_k+1+d'_k)}. \tag{3.26}$$

Let $d'_k \leftarrow d'_k - 1, d'_j \leftarrow d'_j + 1$, the above shows that this does not increase the objective function. Continue this procedure until $d'_k = 0$, then by Case 1 above, (3.22) is obtained.

3. $d'_k < 0$. The proof is similar to the case $d'_k > 0$, except that we use (3.19) instead of (3.20).

Hence Algorithm **Alloc-1-link** terminates with an optimal allocation. ■

Again, note that when $UB_{sd} \neq \infty$ for some $t_{(sd)}$, convexity may be lost in this region. However, this is easy to correct for. First we perform **Alloc-1-link** without representing the nonconvexity, but taking care not to allocate spare capacity to any traffic component beyond its upper bound, which is simple to do. Since for each traffic component the algorithm may have made a "mistake" only in not allocating the final $(UB_{sd} - t)$-th unit of spare capacity, we can make a single pass after completion of the above algorithm and reallocate spare capacity if this will decrease the overall rejection rate. Since spare capacity need only be taken away from the traffic component which causes least rise in rejection rate, this procedure does not cause a combinatorial explosion. We omit details of this straightforward procedure.

For the sake of completeness, we remark that the single link case can be also seen as a flow problem that can be solved by capacity scaling. However, such a view does not provide any guidance to the solution of the general topology case, so we do not discuss it further.

3.5 The General Topology Case

In a realistic traffic grooming network, there will be a logical topology consisting of a large number of links, and end-to-end traffic components will traverse more than one link in general. In this section, we consider this more practical multiple-link problem.

3.5.1 Complexity of the General Case

In multi-hop networks, the same traffic component may traverse more than one logical links. When the traffic component requests an increment, the request can be satisfied only if it has a unit of overprovisioned spare capacity remaining on every link it traverses. Therefore its MTTR is determined by its spare capacity allocation on the link on which it has the smallest spare capacity allocation. Thus larger spare capacity allocated to this component on other links is wasted, but this becomes apparent only when the different links are considered together. Thus coupling is introduced in the problem, and we suspect that the problem becomes conceptually much harder in this case. In this subsection, we show that this is indeed true. First, we present a slightly simpler formulation of the general case.

In the function $m(t,s)$ (more simply, $m(s)$), the only quantity that depends on the link under consideration is the spare capacity. Thus, using the terminology we defined in Section 3.3, it is easy to see that the MTTR M_i of a traffic component i traversing the set of lightpaths $\mathcal{L}_i$ is given by:

$$M_i = \min_{l \in \mathcal{L}_i} \left(m_i(s_i^{(l)}) \right) = m_i \left(\min_{l \in \mathcal{L}_i} s_i^{(l)} \right). \tag{3.27}$$

The overall objective now is to maximize the time until the first rejection of an increment request in the whole network. The MTTR of the whole network is again seen to be the harmonic sum of the MTTRs of the all the traffic components, as a consequence of the result regarding the minimum of exponentially distributed variables referred to in Section 3.3.

Hence the objective function of the general mathematical programming formulation in (3.1) can be obtained as:

$$Q(\{s_i^{(l)}\}) = \frac{1}{\sum_{i=1}^{n} \frac{1}{m_i\left(\min_{l \in \mathcal{L}_i} s_i^{(l)}\right)}} = \frac{1}{z(\{s_i^{(l)}\})}, \tag{3.28}$$

and again, the problem can be posed as one of minimization of the rejection rate $z(\{s_i^{(l)}\})$, now for the whole network.

As we stated above, in any solution, a traffic component may be allocated more spare capacity on some links than others; this allocation is not useful and could be removed without changing the rejection rate achieved by the solution. This is true of optimal solutions also. If we relax the requirements that all the spare capacity on every link must be assigned to some traffic component or other, we can abandon separate variables $s_i^{(l)}$ for the slack allocation received over different links by the same traffic component i, and use a unique variable s_i to represent the spare capacity allocation received by this traffic component over each link it traverses. This allows us to obtain the following simpler formulation:

$$\text{minimize} \quad \sum_i \frac{1}{m_i(s_i)} \tag{3.29}$$

$$\text{subject to} \quad \sum_{i: l \in \mathcal{L}_i} s_i \leq S_l \tag{3.30}$$

$$s_i \in \mathcal{Z}_0^+, \forall i. \tag{3.31}$$

But note that we had to use an inequality in (3.30).

Proposition 6. *The problem of minimizing rejection rate by spare capacity assignment is NP-Hard in a general topology network.*

Proof. Our proof follows a derivation in [**?**], which is a study on the maximum integral multicommodity flow problem. We reduce the NP-Hard three-dimensional matching problem to the spare capacity allocation problem. An instance of the three-dimensional matching problem [**?**] is specified by three disjoint sets $X = \{x_i\}, Y = \{y_i\}, Z = \{z_i\}, i = 1 \ldots n$, and a set of triples $S = \{x_i, y_i, z_i\}$; the problem is to find the maximum number of disjoint triples. We construct an instance of the spare capacity assignment problem as follows. The network is a tree T with with height 3, the root is marked as r. At level 1, there are $3n$ nodes representing the elements in $X \cup Y \cup Z$, labelled as $x_i, y_i, z_i, i = 1 \ldots n$ correspondingly. Between each node x_i and the root, there is a bidirectional link, with 1 unit of spare capacity on each direction. From the root to each node z_i, there is a directed link with 1 unit spare capacity. From each node y_i to the root, there is a directed link with 1 unit spare capacity. Each node $x_i, i = 1 \ldots n$ has p_i children, where p_i is the number of occurrence of x_i in S. These nodes are labelled as $x_{i,l}, 1 \leq l \leq p_i$. There is a bidirectional link between x_i and each $x_{i,l}$, with 1 unit spare capacity on each direction. Each $x_{i,l}$ has two children, left child labelled as $x_{i,l,a}$, right child $x_{i,l,b}$. There is a directed link from $x_{i,l,a}$ to $x_{i,l}$ and a directed link from $x_{i,l}$ to $x_{i,l,b}$, each with 1 unit spare capacity.

Next we define the traffic components. For easy readability, we denote a traffic component from s to d with the notation $(s \rightarrow d)$ or $(d \leftarrow s)$ in what follows. We introduce traffic $(x_{i,l,a} \rightarrow x_{i,l,b})$, $\forall i, l$ and $(x_{i,l,a} \rightarrow z_k)$, $(x_{i,l,b} \leftarrow y_j)$ if $(x_i, y_j, z_k) \in S$. All traffic components have the same existing amount of traffic demand t and arrival rate γ and departure rate μ; thus all have the same MTTR function $m(\cdot)$. An example of this construction is shown in Fig. 3.6, where solid black lines represent logical links and shaded lines represent traffic demand components.

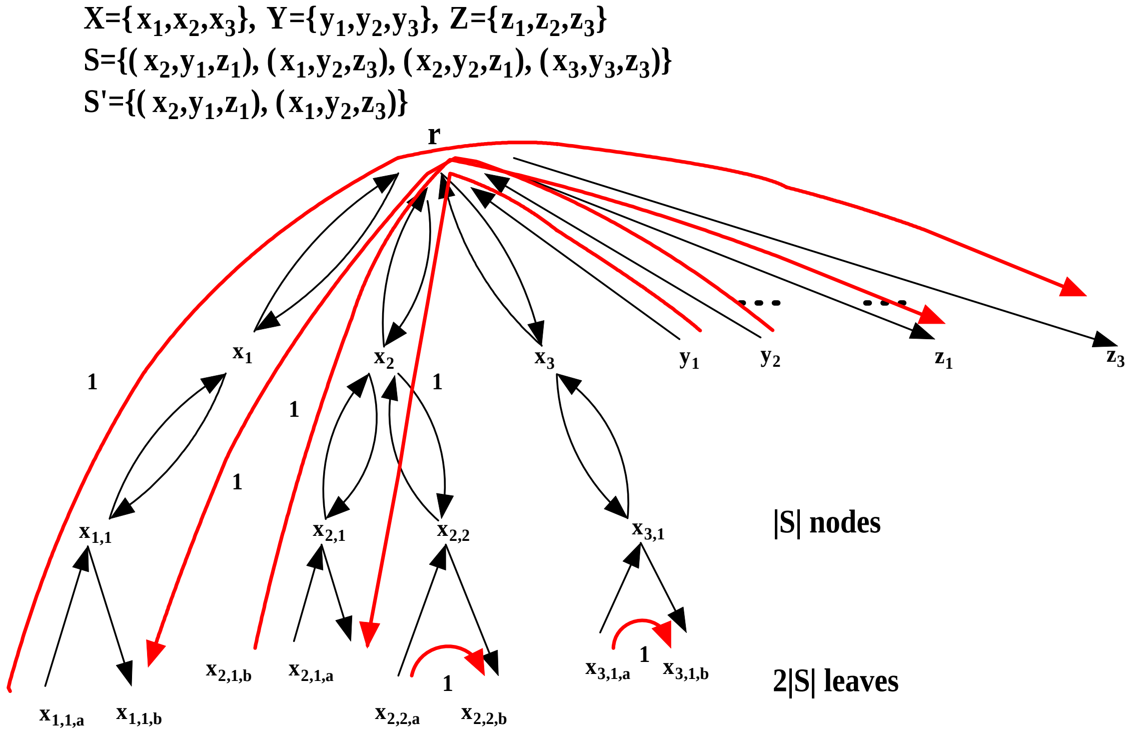


Figure 3.6: Construction of tree network from 3DM

The amount by which the objective function $z(\{s_i^{(l)}\})$ is reduced for one unit of spare capacity assigned to any traffic component is $\triangle = \frac{1}{m(1)} - \frac{1}{m(0)}$. Because the spare capacity available for each traffic component is at most 1 unit, the objective function value is minimized if and only if the maximum amount of spare capacity can be assigned. Now we show that the instance of three-dimensional problem has t disjoint triples if and only if $t+|S|$ units of spare capacity can be assigned in the corresponding spare capacity allocation problem instance constructed. Consider a solution $S' \subseteq S$ to the 3DM problem instance with t disjoint triples. If x_i is not covered by S', we assign 1 unit spare capacity to traffic components from $x_{i,l,a}$ to $x_{i,l,b}$, $\forall 1 \leq l \leq p_i$. Otherwise, if $(x_i, y_j, z_k) \in S'$ and corresponds to the lth occurrence of x_i, then we assign 1 unit spare capacity to the traffic components $(x_{i,l,a} \rightarrow z_k)$ and $(x_{i,l,b} \leftarrow y_j)$ respectively, $\forall m \neq l$. Thus, a covered x_i results in an assignment of $p_i + 1$ unit of spare capacity, and an uncovered x_i results in an assignment of p_i unit of spare capacity. Therefore, the total amount of spare capacity assigned is $t + \sum_{i=1} p_i = t + |S|$.

Conversely, suppose that $t + |S|$ units of spare capacity can be assigned. Note that the maximum amount of spare capacity can be assigned over traffic components that have at least one end in the subtree routed at x_i is $p_i + 1$. Moreover, the assignment can only be done by giving 1 unit spare capacity to traffic components $(x_{i,l,a} \rightarrow z_k)$ and $(x_{i,l,b} \leftarrow y_j)$, for some l, and 1 unit to each of the remaining $p_i - 1$ traffic components, $(x_{i,m,a} \rightarrow x_{i,m,b})$, $\forall m \neq l$. Since $t + |S|$ units of spare capacity are assigned, there must be at least t elements x_i such that $p_i + 1$ units spare capacity are assigned to traffic components that have at least one end in the subtree rooted at x_i. If spare capacity is assigned to traffic components $(x_{i,l,a} \rightarrow z_k)$ and $(x_{i,l,b} \leftarrow y_j)$, then (x_i, y_j, z_k) is a triple in S. Let S' be the set of these (at least t) triples, each x_i, y_j, z_k is included only once because of the amount of spare capacity on the edges between the root and nodes at level 1.

The transformation was carried out in polynomial time, and we have shown that solutions to each problem can be transformed into solutions for the other in polynomial time as well. Since the 3DM problem is NP-Hard, so is the spare capacity allocation problem. ■

In view of the NP-hardness of the problem, in the following subsections, we introduce heuristic algorithms.

3.5.2 Algorithm Alloc-Round: a rounding approach

In this section, we propose flavors of an algorithm that simply rounds optimal solutions obtained for the continuous version, thus this algorithm is akin to Algorithm A1 in Section 3.4.2. We apply nonlinear programming method to find optimal solutions for the continuous optimization problem, then round them to a feasible integer solution. Since we have shown that the continuous version is a separable convex programming problem, it falls into a class of mathematical programming problems that is not *hard* to solve (at least theoretically). However, the result of such rounding is no longer guaranteed to be an optimal solution to the integer version, we merely expect these to be good heuristic approaches.

RR: Randomized rounding. In this approach, we keep a list of all traffic components and round the spare capacity assigned to them down. Then, a traffic component is randomly selected and rounded up if doing so will not violate the link capacity constraint. Remove the variable from the list and keep selecting the next variable until this list is empty.

DR: Derivative based rounding. We first round solution variables down and sort the list according to the derivative at that point (intuitively, this is representative of how much improvement in terms of the objective function value we may obtain if 1 unit spare capacity is assigned to the traffic component), then select the traffic component that has the largest derivative and round it up if doing so will not violate the link capacity constraint. Then, this traffic component is removed from the list. We keep doing so until the list is empty.

WDR: Weighted derivative based rounding (WDR). This method is exactly the same as DR except that the sorting is done according to the weighted derivative, that is, the derivative is divided by the number of links the traffic component traverses. Intuitively, if there are two traffic components having the same derivative (if 1 unit spare capacity is assigned, it will result in the same amount of decrement of objective function value), the "shorter" traffic component (the one traverses a smaller number of links) is a better candidate, because it takes up slack over a smaller number of links.

3.5.3 A Heuristic Based on Relaxation

We develop a heuristic algorithm based on Lagrangian relaxation, as in [**?**]. We use the following notations: $\tilde{\lambda} = \{\lambda_l\}, l = 1, \ldots, m$, where m is the number of links, is the vector of lagrangian multipliers. $\tilde{S} = \{s_i\}, i = 1, \ldots, n$, is the spare capacity allocation vector, where s_i is the number of units of spare capacity assigned to traffic component i.

The algorithm is conceptually in two parts: introducing *fictitious* traffic to remove the inequality constraint of (3.30), and then using the standard technique of Lagrangian relaxation. The heuristic (as opposed to exact) nature of the algorithm lies in the first part; as we show, the duality gap in the Lagrangian relaxation is zero in this case. We also show that relaxing the integrality constraint does not affect the integrality of the solution, for this reason we continue to use the integer spare capacity variables s rather than the continuous variables σ in this section. Thus, the solution obtained from the Lagrangian relaxation will always be an exactly optimal solution to the problem with the fictitious traffic, but may not always be optimal for the original problem.

In (3.29)-(3.31), we simplified the formulation at the cost of introducing an inequality in the place of an equality for the total spare capacity allocated. Effectively, the introduction of fictitious traffic is to regain the advantage of equality. For every link with spare capacity, we add a fictitious traffic component traversing only that single link *if* there is no such traffic component in the given problem instance itself. The current magnitude of the fictitious traffic components are zero and the increment request rates γ are set to a very small value. This ensures that the integral optimal solution to this modified problem must be a solution such that all units of spare capacity are assigned; obviously, in any spare capacity assignment that leaves unassigned units of spare capacity on some link l, we can decrease the objective function value by assigning the unassigned spare capacity to the traffic component that only traverses l. The primal problem is then as follows:

$$\text{minimize} \quad Z(\tilde{S}) = \sum \frac{1}{m_i(s_i)} \tag{3.32}$$

$$\text{subject to} \quad \sum_{i:l \in \mathcal{L}_i} s_i = S_l, \forall l \tag{3.33}$$

$$s_i \geq 0, \forall i \tag{3.34}$$

The lagrangian dual problem is to *maximize* the quantity:

$$\min_{\tilde{\lambda}} L(\tilde{\lambda}) = -\sum_i \frac{1}{m_i(s_i)} + \sum_l \lambda_l \left(\sum_{i:l\in\mathcal{L}_i} s_i - S_l \right) \tag{3.35}$$

We can neglect the non-negativity constraints for variables $s_i, i = 1, \cdots, n$, by incorporating them into the cost function, i.e., setting the value of the cost function as infinity when an s_i reaches 0^-. In solving the dual, we can also ignore the integrality constraints on spare capacity variables; as we see below in Figure 3.7 and the discussion following, an integer solution will always be obtained if the piecewise linear representations (conceptually defined on the continuous domain) are used. It is easily shown that the duality gap for this Lagrangian relaxation is zero. Briefly, suppose we have a primal feasible solution $\tilde{S}^*$ that is optimal to the primal problem (3.32)-(3.34), and a dual solution $\tilde{\lambda}^*$ that is optimal to the lagrangian dual problem (3.35). Since the second term of equation (3.35) is always zero, $Z(\tilde{S}^*) = L(\tilde{\lambda}^*)$.

It is shown in [**?**] that a primal feasible solution $\tilde{S}^*$ satisfying constraint (3.33) is optimal if and only if there exists $\tilde{\lambda}$ such that $\sum_{l:l\in\mathcal{L}_i} \lambda_l$ is sub-gradient at s_i^*, for each traffic component i. Equivalently, it is optimal if and only if:

$$s_i^* = \text{argsup}_{s_i} \left(\sum_{l:l\in\mathcal{L}_i} \lambda_l s_i - \frac{1}{m_i(s_i)} \right) \ \forall i \tag{3.36}$$

is satisfied. Fig. 3.7 illustrates the nature of this function. From the figure, because $-\frac{1}{m_i(s_i)}$ is an increasing function, $\sum_{l:l\in\mathcal{L}_i} \lambda_l s_i - \frac{1}{m_i(s_i)}$ is unbounded if $\sum_{l:l\in\mathcal{L}_i} \lambda_l \geq 0$, and it attains the unique supremum at $s_i^* : \sum_{l:l\in\mathcal{L}_i} \lambda_l \in \vartheta\left(\frac{1}{m_i(s_i^*)}\right)$, if $\sum_{l:l\in\mathcal{L}_i} \lambda_l < 0$, where $\vartheta\left(\frac{1}{m_i(s_i^*)}\right)$ is the set of sub-gradient of function $\frac{1}{m_i(s_i)}$ at s_i^*. Since $\frac{1}{m_i(s_i)}$ is monotonically decreasing, given the vector $\tilde{\lambda}$, $\tilde{S}$ can be easily obtained by searching techniques. Moreover, Fig. 3.7 shows that s_i is an increasing function of $\sum_{l:l\in\mathcal{L}_i} \lambda_l$ because as $\sum_{l:l\in\mathcal{L}_i} \lambda_l$ gets smaller, the supremum moves towards 0. This also allows us to observe that the best integer solution (better than neighboring integer solutions) must necessarily be better than non-integer solutions in the near vicinity, because of the piecewise linear nature of the function. Thus integer solutions will be obtained even if integrality constraints are not explicitly enforced, as we remarked above.

We define infeasibility in the dual as follows. Since $\frac{\partial L(\tilde{\lambda})}{\partial \lambda_l} = \sum_{i:l\in\mathcal{L}_i} s_i - S_l, \forall l$, if the solution is primal feasible, we have $\frac{\partial L(\tilde{\lambda})}{\partial \lambda_l} = 0, \forall l$. We call $|\beta| = |\frac{\partial L(\tilde{\lambda})}{\partial \lambda_l}|$ the infeasibility of the constraint on link l, if $\beta \neq 0$.

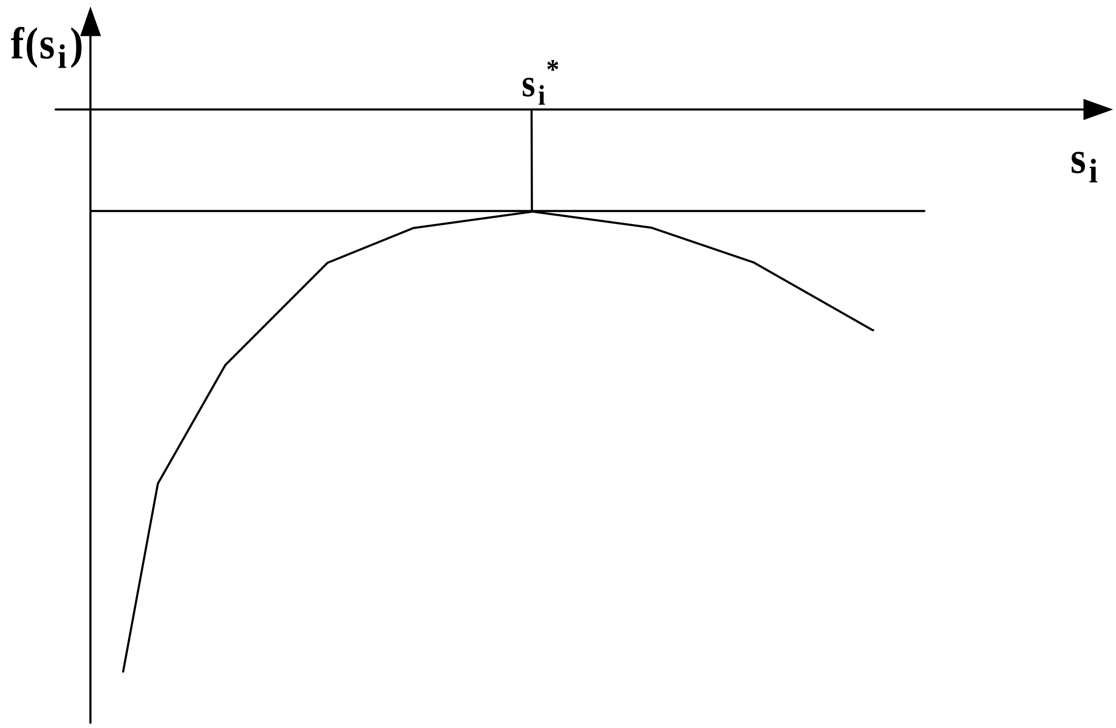


Figure 3.7: An illustration of the function

We are now ready to describe the relaxation algorithm. The superscript r is used to represent the variables at the start of the r-th iteration and $\delta \in (0, 1)$ is a predefined real number. In our implementation, we choose a coordinate (traffic component) c in a cyclical manner in the iterations, and $\triangle_c$ is found by binary search.

Algorithm **Alloc-Relax**:

1. repeat
2. find $\tilde{S}^r : s_i^r = arg\, \sup_{s_i} \left(\sum_{l:l \in \mathcal{L}_i} \lambda_l^{r-1} s_i - \frac{1}{m_i(s_i)} \right)\ \forall i$
3. compute $\frac{\partial L(\tilde{\lambda})}{\partial \lambda_l} : \frac{\partial L(\tilde{\lambda})}{\partial \lambda_l} = \sum_{i:l \in \mathcal{L}_i} s_i^r - S_l, \forall i$
4. choose any coordinate c, set $\beta = \frac{\partial L(\tilde{\lambda})}{\partial \lambda_c}$
   (a) if $\beta = 0$, do nothing
   (b) if $\beta > 0$, $\lambda_c^r = \lambda_c^{r-1} - \triangle_c : 0 \leq \frac{\partial L(\tilde{\lambda})}{\partial \lambda_c} \leq \delta\beta$

(c) if $\beta < 0$, $\lambda_c^r = \lambda_c^{r-1} + \triangle_c : 0 \geq \frac{\partial L(\tilde{\lambda})}{\partial \lambda_c} \geq \delta\beta$

5. until gain in objective stabilizes, OR maximum number of iterations is exceeded

In each iteration of the algorithm, the optimality condition (3.36) is maintained, and the infeasibility β is reduced by updating $\tilde{\lambda}$. It is always possible to so reduce the infeasibility because of the following reason. Suppose for a chosen coordinate c, $\beta > 0$ (respectively, $\beta < 0$), λ_c^{r-1} is decreased by $\triangle_c$ (respectively, increased by $\triangle_c$); since $s_i^{r+1} \forall i : c \in \mathcal{L}_i$ is increasing function of λ_c^r, the infeasibility $|\beta|$ is also reduced.

Next, we briefly analyze the effect of fictitious traffic components. The basic idea is that by adding fictitious traffic components, we can turn the optimization problem with inequalities to an optimization problem with equalities, which is conceptually easier to solve. We found that after relaxing the equality constraints, the integer and non-negativity constraints of the primal problem are redundant because the optimal solution of the Lagrangian dual problem is always integral and non-negative for a given dual vector. Note that this is different from adding slack variables to the original problem because the slack variables have non-negativity constraints, which cannot be omitted in the dual problem. Moreover, the dual variables are free (with no requirements on their signs), thus, the Lagrangian dual problem is an unconstrained problem, which is easy to solve.

However, by adding fictitious traffic components, the original problem has been modified. The difference is interesting in terms of the well-known Karush-Kuhn-Tucker (KKT) conditions. In the original problem, we have the complementary slackness condition, that is, if on a link, the optimal solution does not claim all the spare capacity (the sum of spare capacity assigned to traffic components traversing the link is strictly less than the total amount of spare capacity, suppose the amount of unused capacity is s_u), the corresponding optimal dual variable of the link must be zero. However, after adding fictitious traffic components, because of the equality constraints, the complementary slackness condition is always satisfied. Now, suppose that the optimal solution to the original problem is also optimal to the modified problem. It turns out that the dual variable on link l must also be a sub-gradient of the cost function of the fictitious traffic component on link l at point s_u. Since that the arrival rate of the fictitious traffic component is very low, the sub-gradient at s_u is an interval very close to zero. Therefore, instead of being zero in the original problem, the dual variable is relaxed to be in an interval very close to zero if fictitious traffic components are added.

3.5.4 Algorithm Alloc-Greedy: greedy heuristics

The above algorithm has good performance but is somewhat complex. In some cases, it may be suitable for implementation for on-line application in the network. In other cases, a simpler approach may be desired. For such cases, we propose three different flavors of a very simple greedy approach in this section.

GA: First, we propose the basic greedy approach. We use the optimal algorithm developed for the single-link model on each link. Because a traffic component can be assigned different amounts of spare capacity on different links it traverses, we pick the minimum amount and assign this amount of spare capacity to the traffic component on every link it traverses. After doing this for all traffic components, we may have some unused capacity left on each link. We simply assign it to single-link traffic if it exists. The algorithm is as follows:

1. Let S_l be the available spare capacity on link l, use Algorithm **Alloc-1-link** for each link
2. Let $\mathcal{P} = \{i : |\mathcal{L}_i| > 1\}$, for each i do
3. $\quad s_i = s_i^{(k)} : k = \text{argmin}_{l \in \mathcal{L}_i} \frac{1}{m_i^{(l)}(s_i^{(l)})}$,
4. $\quad S_l \leftarrow S_l - s_i^{(k)}, \forall l \in \mathcal{L}_i$
5. For each l, allocate S_l to all traffic components traversing l and $\notin \mathcal{P}$ using Algorithm **Alloc-1-link**.

This algorithm is heuristic in the sense that a traffic component can lend its over-allocated spare capacity to other single-hop traffic components, since this will benefit others without sacrificing itself. However, it is possible that borrowing some spare capacity from other traffic components on the critical link, instead of lending over-allocated spare capacity to other traffic components on other links, is more beneficial. That is, the amount of spare capacity allocated on the critical link is not necessarily a lower bound of the optimal allocation of the traffic component. In what follows, we will discuss a modified version that involves a more complicated lending procedure.

IA: Iterative Algorithm. This algorithm is a modification of the previous one.

1. Initialize $s_i = 0 \forall i$

2. while $\exists l : S_l > 0, \forall l$
3. let S_l be the available spare capacity on link l, use Algorithm **Alloc-1-link** for each link
4. let $\mathcal{P} = \{i : |\mathcal{L}_i| > 1\}$, for each i do
5. $\quad s_i = s_i + s_i^{(k)} : k = \text{argmin}_{l \in \mathcal{L}_i} \frac{1}{m_i^{(l)}(s_i^{(l)})}$,
6. $\quad S_l \leftarrow S_l - s_i^{(k)}, \forall l \in \mathcal{L}_i$
7. end while

The basic idea is that, using the algorithm for the single link case, we may assign different amount of spare capacity on different links for a traffic component traversing more than one link. However, instead of simply assigning the traffic component the minimum amount of assignment over all links and giving the spared capacity (if exists) to single link traffic components, we use an iterative approach. At each iteration, this modified version reserves the minimum assignment for the traffic component, that is, removes that amount from the spare capacity, then repeats the single link algorithm. The iteration stops when no improvement can be made. Obviously, this algorithm is more complicated than the previous algorithm because, after an amount of spare capacity is reserved for a traffic component, it can still take part in the competition of gaining more spare capacity.

WIA: Weighted Iterative Algorithm. This is a length-weighted variation to the previous iterative algorithm. Consider the case that, when we use the greedy approach on a single link, we greedily pick a traffic component that provides the largest reduction of the objective function value if 1 unit spare capacity is assigned. In the multi-link case, when two traffic components traversing a link provide the same amount of reduction, we break the tie by choosing the shorter one so that the saved spare capacity on some link can be assigned to other traffic components. In this heuristic, we formalize that intuition by defining a *weighted reduction* of the objective function for each traffic component, as the number of the links it traverses divided by the actual reduction of the objective function value if 1 unit spare capacity is assigned. This provides us a length-weighted approach.

3.6 Numerical results

For the single logical link case, we compare the results of algorithm **Alloc-1-link** with a brute-force algorithm that finds the optimal by enumerating all the possibilities. The numerical result verifies that the algorithm is optimal as we have concluded. This is not unexpected, and we do not produce any of those results here.

For the general topology, we generate random logical topologies as follows.

1. Input number of nodes n, number of arcs m and k, which satisfy $n + 2k \leq m$.

2. Put vertices randomly in k sets, $V_1, \ldots, V_k$ with cardinalities $n_1, \ldots, n_k$. Randomly select unassigned nodes and assign to the different sets in rotation.

3. Within each subset V_j, randomly permute the members, then add arcs to create a directed circuit in the random order obtained.

4. Within each set V_j, randomly select two nodes a_i and b_i. Add arcs $(a_1, b_2), (a_2, b_3), \ldots, (a_{k-1}, b_k)$.

5. Randomly select pairs of nodes and add the arc between them if not already in the network until the total number of arcs is m.

The capacity of each link C is set to 16. We also generate traffic matrices with sub-wavelength traffic components. The shortest path routing algorithm is used to route the traffic matrix on the logical topology. For the sake of simplicity, the traffic matrices are generated such that using the shortest path routing will not violate the capacity constraint of the link. Specifically, we generate sub-wavelength traffic demands randomly in the interval $[1, 16]$. If a link capacity is violated, all the traffic components traversing the link are regenerated from an interval $[1, 15]$, etc. We apply all algorithms on each randomly generated instance. For each instance, each algorithm generates an assignment of the spare capacity on each link. The MTTR of the network is computed according to the assignment.

To obtain a baseline solution for each instance generated, we solve the continuous version to find the *optimal continuous solution*. This serves as a lower bound on the optimal integer solution. Ideally we should compare the performance of our algorithms with the optimal integer solution, but it turned out to be computationally impossible to obtain actual numerical results for the optimal. Because of the convexity property, the overall problem can be solved in reasonable time by standard nonlinear programming methods, as long as the integer constraints are relaxed. Some instances of the results are shown in Table 1. The significance of the columns are as follows. The first column shows the lower bound obtained from the continuous solution. The rest of the columns show the results obtained using the relaxation method and the greedy heuristics. For easy comparison of the results with the lower bound provided by the continuous solution, we show both the absolute values of the solutions produced by the different heuristics as well as the same values normalized to the lower bound.

We generated hundreds of instances and plot the results in the following figures according to different view points. Only the results normalized to the lower bound are plotted. All the figures use MATLAB^{TM} boxplots, which allow the representation of a large amount of data concisely. Briefly, each box represents a population of data, the lines indicate the lower quartile, median, and upper quartile values. The long protruding lines show the extent of data, with '+' indicating outliers. More details can be found in [71].

First, we study networks with different sizes. Fig.3.8 and 3.9 show the results on randomly generated networks with $5 - 10$ and $11 - 16$ nodes respectively. Secondly, we study networks with different maximum r, which reflects how fast traffic changes, in all traffic components. Fig.3.10 and 3.11 show the results on networks with maximum $r \geq 20$ and < 20 respectively. Thirdly, we study networks with different minimum r in all traffic components. Fig.3.12 and 3.13 show the results on networks with minimum $r \geq .2$ and $< .2$ respectively. Finally, we study networks with different maximum load in all links. Fig.3.14 and 3.15 show the results on networks with maximum load $\geq$ OC-20 and $<$ OC-20 respectively.

Our computational experience shows that in general, the relaxation algorithm runs slower than other heuristics, but faster than finding the continuous optimal. The main reason is that, using the piecewise linear objective function, for a given dual vector, the optimal $s_i, i = 1, \ldots, n$ can be easily found in logarithmic time in terms of the number of pieces. Moreover, the solution generated by the relaxation algorithm is generally better than that of other approaches. However, in our implementation, we have a maximum number of iterations. If the program stops before it reaches the maximum number of iterations, it generates a feasible solution because it satisfies all the primal constraints. In addition, the Lagrangian dual problem is always a lower bound of the primal problem, thus, this solution is also optimal to the primal. However, we find that in some cases, the algorithm does not quit until the maximum number of iterations is reached. In these cases, the solutions may

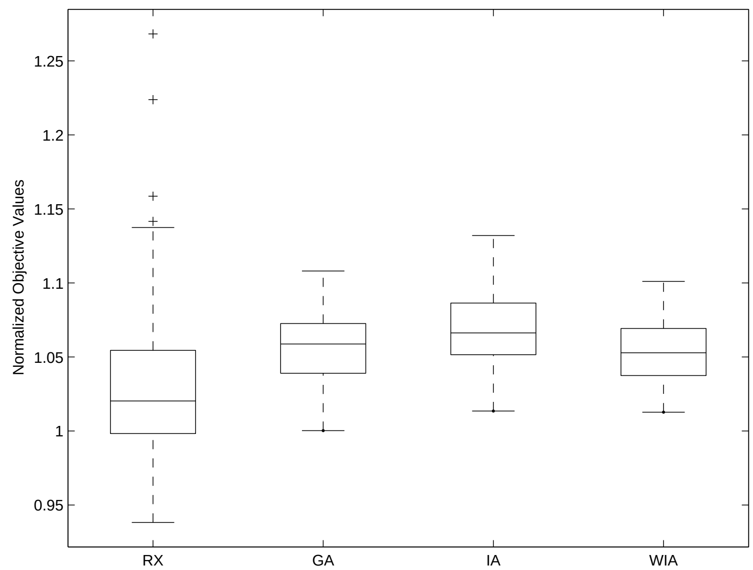


Figure 3.8: Numerical results: small networks (N$\in [5, 10]$)

be infeasible to the primal. This is the case for the instances marked with a '*' in Table 1, where the solution obtained appears to be better than the lower bound. Such solutions would need some adjustments to make them feasible, which shows an advantage of the straightforward greedy approaches. The rounding methods perform well, with the weighted derivative method outperforming the derivative based methods, as expected. The greedy approaches in general perform very well, and usually the best greedy solution outperforms all the rounding approaches. This is a rather surprising but practically very welcome result. In general, the iterative approaches work better than the simple approach. Since the greedy algorithms are computationally cheap, a practical strategy might be to run them all and adopt the one providing best results.

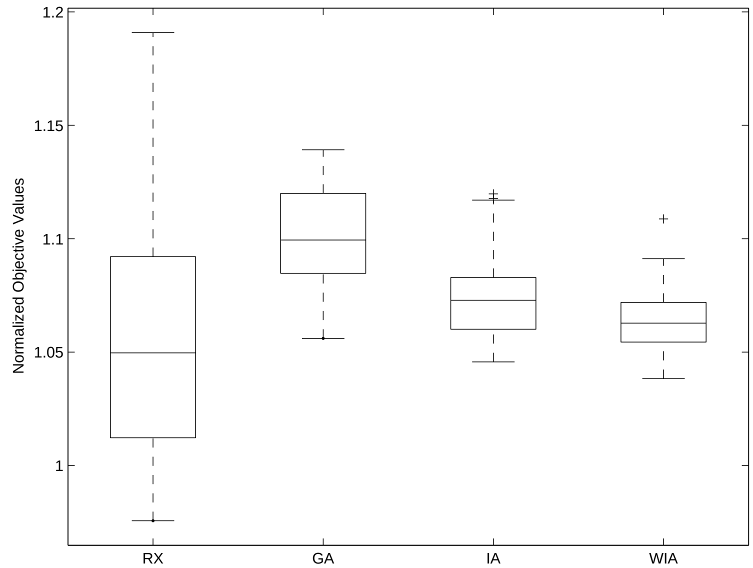


Figure 3.9: Numerical results: larger networks (N$\in [11, 16]$)

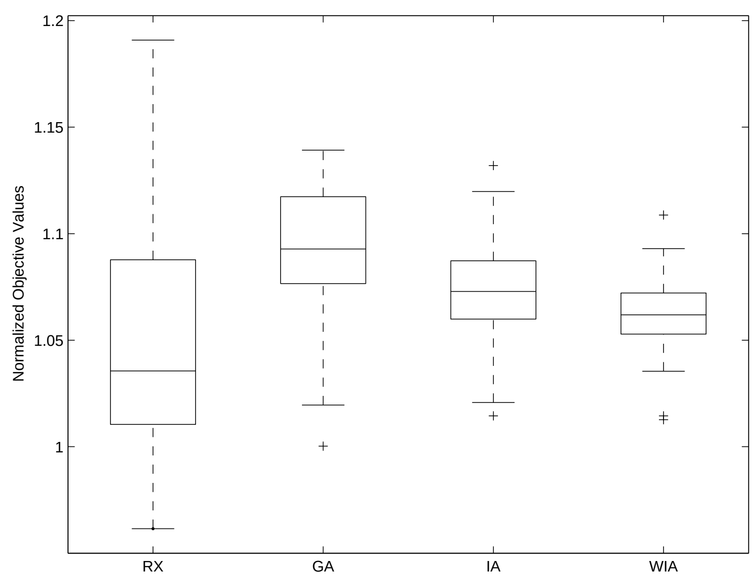


Figure 3.10: Numerical results: maximum $r \geq 20$

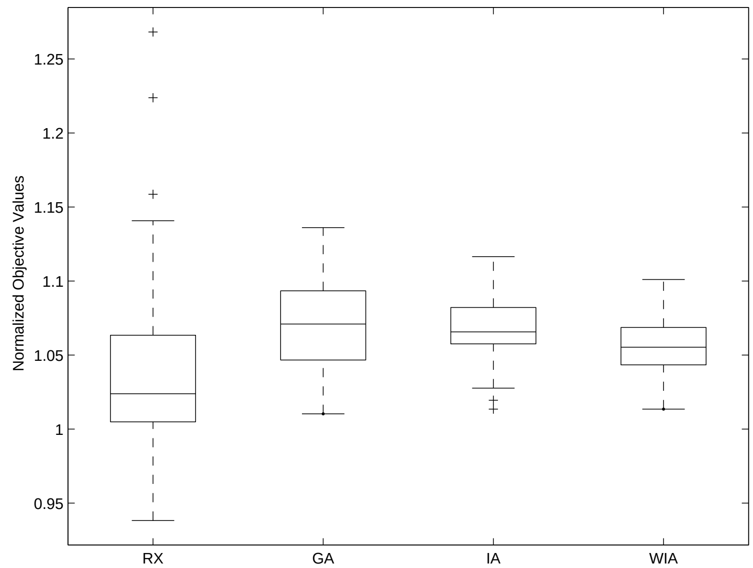


Figure 3.11: Numerical results: maximum $r < 20$

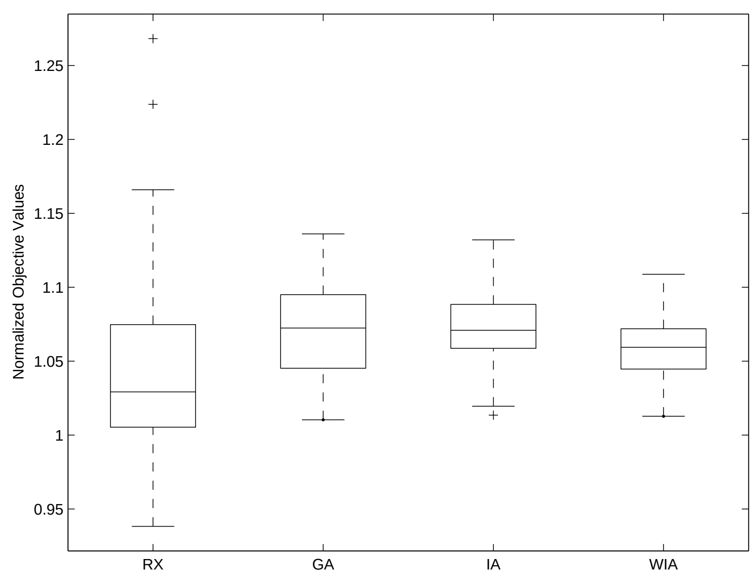


Figure 3.12: Numerical results: minimum $r \geq .2$

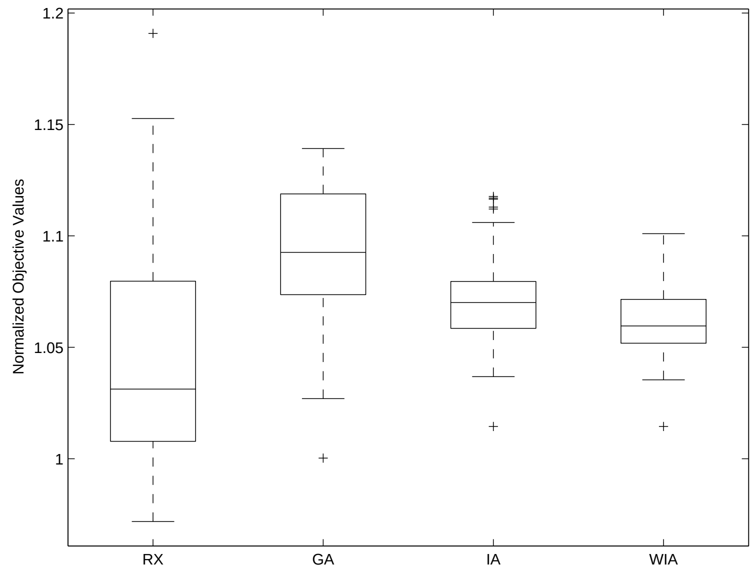


Figure 3.13: Numerical results: minimum $r < .2$

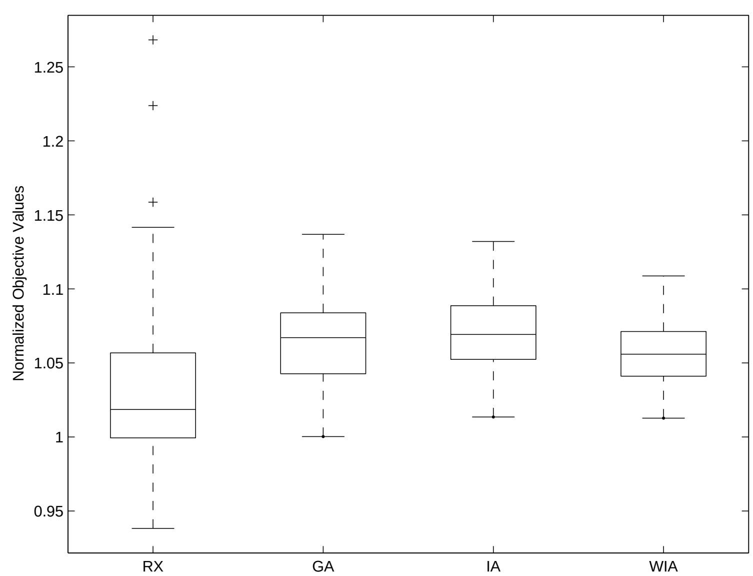


Figure 3.14: Numerical results: maximum load $\geq$ OC-20

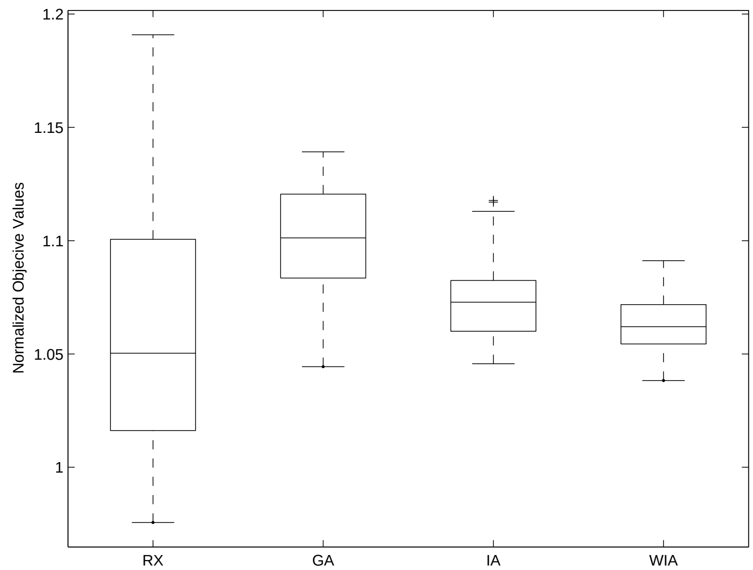


Figure 3.15: Numerical results: maximum load < OC-20

Table 3.1: Rejection rates obtained for randomly generated sample of instances, all algorithms

Cont.	RX	RX/Con	DR	DR/Con	WDR	WDR/Con
23.54101	23.90313	1.015383	25.867136	1.098811	25.377573	1.078015
35.98379	40.80513	1.133987	40.175617	1.116492	41.250645	1.146367
25.74975	25.71163	.998520*	27.97374	1.086369	28.856234	1.120641
105.5301	118.8668	1.126378	125.699	1.191119	121.00771	1.146665
24.16574	23.91263	.989526*	26.423716	1.093437	27.259634	1.128028
99.38826	107.6342	1.082967	115.89671	1.1661	115.77002	1.164826
51.34201	50.12341	.976265*	57.651546	1.122892	57.922066	1.128161
17.10927	17.06912	.997653*	18.432013	1.077311	18.642866	1.089635
34.36739	35.60514	1.036015	38.654366	1.12474	39.186367	1.140219
7.942096	8.079233	1.017267	8.310723	1.046414	8.089975	1.01862

RR	RR/Con	GA	GA/Con	IA	IA/Con	WIA	WIA/Con
25.38518	1.07833	25.8187	1.09675	25.4745	1.08213	24.92648	1.05885
40.52795	1.12628	38.8030	1.07834	37.6272	1.04567	37.53914	1.04322
27.36318	1.06265	26.8931	1.04440	27.2991	1.06017	26.84066	1.04236
125.4081	1.18836	119.760	1.13484	113.411	1.07468	113.4113	1.07468
26.97327	1.11617	25.6612	1.06188	25.9375	1.07331	25.53633	1.05671
111.9546	1.12643	109.373	1.10046	105.344	1.05992	105.3445	1.05992
57.39315	1.11785	55.5841	1.08262	54.4784	1.06108	54.07406	1.05321
18.23173	1.06560	18.1151	1.05879	18.5190	1.08239	18.17491	1.06228
38.77751	1.12832	37.8448	1.10118	37.5745	1.09332	36.68803	1.06752
8.31553	1.04702	8.28016	1.04256	8.16850	1.02850	8.09509	1.01926

Part III

Wireless Networks

Chapter 4

Wireless Mesh Networks

Wireless Mesh Networks (WMNs) are mesh networks implemented over Wireless LANs. In a WMN, the wireless nodes, which can be laptops, sensors, etc., form a mesh topology where each node is equipped with one or multiple radios that can transmit/receive/forward packets to other nodes. The following figure 4.1 shows an example of the WMN.

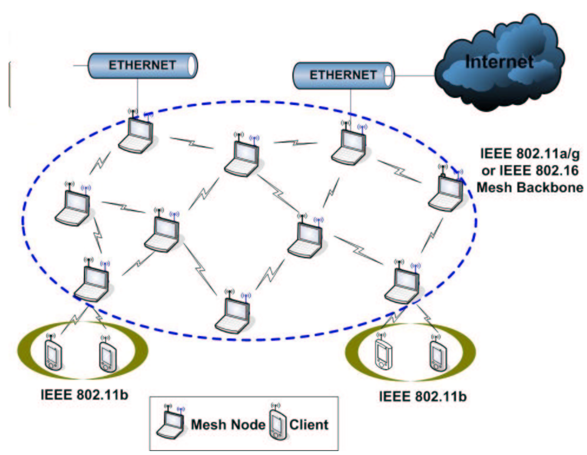


Figure 4.1: An illustration of Wireless Mesh Networks

The design of WMNs is a new field of network design that has seen growing interest in the last few years. Wireless mesh networks are a new arena for multihop wireless networking that is confidently expected to be an important entity in the campus- or even metro-scale access networks. These networks exhibit characteristics that are novel in the wireless context, and in many ways more similar to traditional wired networks. Because mesh networks are intended to form a backbone over a significant area, node deployment is planned. Nodes are not mobile, and link quality is reasonably stable, with dependable bandwidth over time. Traffic demands made on the network also are reasonably large, and aggregates are predictable in the same sense as in traditional networks of this scale. These networks thus present a very different picture from mobile ad-hoc wireless networks, and are amenable to network resource design methodologies as traditionally applied to planned network deployments.

However, the one wireless reality that must be accommodated in design is that of interference. This is one of the significant differences between WMNs and optical networks. Indeed, in optical networks, similar interference does exist. However, optical filters are well designed such the interference between different channels obeying the standard is generally ignored in the network design. Unfortunately, this is not ture for WMNs where nodes are normally equipped with inexpensive radios. It is, of course, possible to assume a CDMA-like environment, and ignore radio interference. However, in the short term, many wireless mesh environments are likely to be built utilizing 802.11 technology and similar, where such an assumption is not realistic. Further, with mesh applications, as opposed to cell telephony where every user node may be expected to communicate only with the tower, the use of multiple radios at each node is fully expected. In such cases, two radios at the same node may not be possible to use simultaneously to transmit and receive because of near-channel interference, if the channels in use are the same or close together. Indeed, it has been reported that a laptop equipped with two wireless cards using two near channels simultaneously may result in a significant slow-down of file transfering speed. Finally, even transmissions involving completely different node pairs may interfere with each other because of their physical location and the radio medium. The design problem must be formulated with explicit representations of these constraints.

Generally speaking, the design problem can be seen to consist of three subproblems: 1) Routing problem; *i.e.*, how the traffic demands are routed on the multi-hop network. 2) Channel assignment problem; this problem consists of two parts: the connection-radio assignment of radios to connections and the radio-channel assignment of channels to radios. 3) Scheduling problem; since different connections may share the same wireless link, they need to be scheduled. Realistically, the sharing of the link is achieved by time-slotting. Therefore, this subproblem is to assign time slots to connections, on each link.

4.1 Prior Work

In the context of WMNs, lots of researches have been conducted. In the following table, they present their differences. Some researchers study single channel WMNs, while other researchers consider mulitple channels. However, to the best of our knowledge, all of them ignore the near-channels interference. While most researchers consider the general case where there are multiple traffic demands, in [72, 73], the authors also study the special case where a single traffic demand exists. The objectives of the researches are also different. In [74, 75], the objective of interest is to maximize the goodput of the network. In [1, 76], the authors formulate it as a maximum concurrent flow problem where a common scalar for all traffic demands is maximized. The authors of [77] focus on the minimization of the swiching cost because radio adjustments are undesirable. The optimization goal of [78] is to minimize the delay, which is expressed in terms of the Expected Transmission Time (ETT). In [72], Kodialam and Nandagopal study the feasibility problem, that is, if the network can satisfy the traffic demands. In [73], in order to satisfy the traffic demands, some bounds on the resource are derived. In [72], the authors study the routing and channel assignment problems based on the protocol interference model (to be discussed in section 4.4). In [1], they extend the work to multi-channel networks and take the scheduling problem into consideration. The authors propose an initial interference model and claim that this model is equivalent to a somewhat simpler model, which is used to derive the main conclusions. However, we find this claim to be incorrect; the initial model is in fact overly restricted. We discuss this further in Section 4.4. However, since the approach in that work is based on the later formulation, this over-restriction does not affect the correctness of any other result of [1]. In [76], the authors studied the same problem as in [1]. The difference is that, in [76] (also in [75], the radios are static, that is, not allowed to be adjusted during the operation time of the network. In [73], the authors also present an ILP formulation for the routing and scheduling problems with an objective to maximize the throughput. For both the protocol interference model and the physical interference model (to be discussed in section 4.4), *conflict graphs* are constructed. However, in the physical model, the weights of edges in the conflict graph are not linear. In [74, 78, 75], the authors study the case that each node is equipped with two radios and show significant improvement of performance can be achieved over the single radio case.

Table 4.1: Wireless Mesh Networks Literature Review

	# of Ch.	# of SDs	Obj.	Rout.	CA	Sch.	Radio	# of Rad.
[74]	M	M	Goodput	×	×			2
[1]	M	M	Scalar	×	×	×	Dynamic	General
[77]	M	M	Switching	×	×		Both	General
[78]	M	M	ETT	×				2
[72]	S	Both	Feasibility	×		×		
[73]	S	Both	Bounds	×		×		
[75]	M	M	Goodput	×	×		Static	2
[76]	M	M	Scalar	×	×	×	Static	General

4.2 Our Contribution

The formulations for this problem available in the literature may largely be classified as "symmetric". The physical additive interference model, so far not considered in this literature, is not amenable to the symmetric formulation. We provide an asymmetric formulation that, while more complex, is still linear and reflects this interference model accurately. Since the additive interference model is not amenable to symmetric formulations, it is not adequately addressed by the approaches available in the literature. We present an approach that can be used to obtain solutions for the network design problem under this interference model. Our approach is based on the insight that one part of the problem is essentially a generalized matching problem, and iterative use of blossom inequalities can be used to obtain solutions. For the other parts of the problem, we use standard approaches. We obtain numerical results by generating random networks and applying our algorithm. The results show that our approach is of practical value.

The rest of this chapter is organized as follows. In the next two sections, we discuss the different ways to formulate the network design problem for WMNs. Section 4.3 discusses formulation of all aspects of the problem *other than* interference, and Section 4.4

discusses formulations of interference constraints. In Section 4.5 we present our approach to obtaining a solution to the design problem. Section 4.6 presents numerical results.

4.3 Integer Linear Program Formulations

In the following subsections, we present an *edge-based* ILP formulation of the problem, followed by a *node-based* one. That is, the variables of interest are defined on links in the first case, nodes in the other. All aspects of the problem *other than radio interference* are represented in these formulations. As we shall see in the next section, the same choice presents itself in formulating interference constraints, so we need to have these alternate formulations for the rest of the problem available.

Formulations for the protocol model (discussed below) are available in the literature, but we define our own below, because we introduce the variables that allow us to go on to formulations of the additive model for which we contribute a new formulation that we can use in our solution approach. The problem is decomposed into the routing problem, channel assignment problem and the scheduling problem. The constraints are ordered by these subproblems. This formulation is similar to the general multi-commodity flow problem in networks. We do not list any specific objective function, this general formulation can carry over easily to different network design objectives.

4.3.1 An edge-based problem formulation

We assume that the traffic demands are given as a traffic matrix $\Lambda = [\lambda_{ij}]$, where λ_{ij} is the traffic demands (say, in Mbits/sec) from node i to node j. As in [1], the physical topology is $G(V, A, A^I)$, where V is the set of wireless nodes, A is the set of wireless data links and A^I is the set of interference links. Two nodes are connected by a data link if the receiver is within the communication range of the sender. Two nodes are connected by a interference link if the receiver is out of the communication range but within the interference range of the sender, *i.e.*, a transmission by the sender will be perceived as significantly heightened noise at the receiver, thus degrading the Sigal-to-Noise Ratio (SNR) for (therefore, interfering with) a transmission from some other nodes that the receiver might be trying to receive. All links are directed. Let the number of channels be N, the number of radios at node a be

$K(a)$ and the number of time slots be T. Here, the concept of time slots is used in the same manner as a TDM slotted medium. The problem posed is that of assigning the requisite amount of data transfer to each traffic demand component over the entire T slots. Let C be the capacity of a channel, in the same units as the traffic matrix components. We need to find the routing, channel assignment and the scheduling.

We define the following variables. Let $r_{ab}^{(ij)}$ be the traffic flow from i to j and traversing link ab, $r_{ab,n}^{(ij)}$ be the traffic flow from i to j traversing ab and using channel n. Let $l_{n,t}^{(ab)}$ be binary, which is 1 if the channel n on link ab is used in time slot t. Then, we have the following constraints.

- Routing constraint:

$$\sum_{b:ab\in A} r_{ab}^{(ij)} - \sum_{b:ba\in A} r_{ba}^{(ij)} = \begin{cases} \lambda_{ij} & \text{if } a = i \\ -\lambda_{ij} & \text{if } a = j \\ 0 & \text{otherwise} \end{cases}$$

$$\forall ab \in A, \forall i, j \tag{4.1}$$

- Channel assignment:

$$\sum_{n} r_{ab,n}^{(ij)} = r_{ab}^{(ij)} \quad \forall ab \in A, ij \tag{4.2}$$

- Scheduling:

$$\sum_{ij} r_{ab,n}^{(ij)} \leq \frac{C}{T} \sum_{t} l_{n,t}^{(ab)} \forall ab \in A, n, t \tag{4.3}$$

$$\sum_{b':ab'\in A} \sum_{n} l_{n,t}^{(ab')} + \sum_{b':b'a\in A} \sum_{n} l_{n,t}^{(b'a)} \leq K(a) \forall a \in V, t \tag{4.4}$$

Constraint (4.3.1) ensures the flow conservation. Constraint (4.2) ensures that the flow is distributed to channels. The scheduling constraints ensure that the time slots are assigned to flow such that the channel capacity and radio availability are respected.

4.3.2 A node-based formulation

The edge-based formulation presented above is more common in relevant literature. It maps the traffic flow to channels directly and thus hides the detail of the radio assignment. However, the capability of describing more complicated cases (*e.g.*, nodes equipped with heterogeneous radios) is lost in consequence. A more explicit option is the node-based formulation that defines variables for radios at each node. As we show later, this model can also naturally apply to the physical interference model. We present such a model below.

We define additional variables as follows. Let $r_{ab}^{(ij)}$ be the traffic flow from i to j and traversing ab, $r_{ab,k,g}^{(ij)}$ be the traffic flow from i to j, traversing ab and using radio k at a, g at b. $s_{k,n,t}^{(a)}$ is a binary, which is 1 if the radio k at node a is transmitting on channel n at time slot t, $t_{k,n,t}^{(a)}$ is a binary, which is 1 if the radio k at node a is receiving on channel n at time slot t, $l_{k,g,n,t}^{(ab)}$ is a binary, which is 1 if the radio k at node a is transmitting to radio g at node b on channel n over time slot t. Now we have the following constraints:

- Routing constraint: same as constraint (4.3.1).
- Connection-Radio assignment

$$\sum_k \sum_g r_{ab,k,g}^{(ij)} = r_{ab}^{(ij)} \forall ab \in A, ij \tag{4.5}$$

- Radio-Channel assignment and scheduling

$$\sum_{ij} r_{ab,k,g}^{(ij)} \leq \frac{C}{T} \sum_n \sum_t l_{k,g,n,t}^{(ab)} \forall ab \in A, k, n, t \tag{4.6}$$

$$\sum_n \left(s_{k,n,t}^{(a)} + t_{k,n,t}^{(a)} \right) \leq 1 \forall a \in V, k, t \tag{4.7}$$

$$\sum_{b \in V} \sum_g l_{k,g,n,t}^{(ab)} \leq s_{k,n,t}^{(a)} \forall ab \in A, k, n, t \tag{4.8}$$

$$\sum_{a \in V} \sum_k l_{k,g,n,t}^{(ab)} \leq t_{k,n,t}^{(b)} \forall ab \in A, g, n, t \tag{4.9}$$

Constraint (4.5) ensures the flow from i to j traversing ab is assigned with interface(s). Constraint (4.6) ensures the capacity of the link is respected. Constraint (4.7) ensures a radio can be either transmitting or receiving, but not both, on a channel in any time slot. Constraint (4.8) and (4.9) ensure that a radio can communicate with at most one radio. In this, we have made the assumption, usual in the WMN context, that most traffic is unicast, and utilization of the broadcast nature of the medium for multicast traffic is not useful.

4.4 Formulations of Interference Models

In wireless networks, radio interference is an essential characteristic. In [79], Gupta and Kumar propose two interference models, one is a protocol interference model, the other is a physical model. The protocol interference model simulates the MAC layer protocol, i.e., the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) machanism with Ready To Send/Clear To Send (RTS/CTS) exchanges. The physical interference model studies the Signal Noise Ration (SNR) at each node directly. It is more flexible because it can take antenna gain, background noise and some other issues into consideration. We will provide more details in section 4.4.3. To clarify the consequences of different decisions in formulating this aspect of the problem, we first discuss the issues to consider in deciding between alternative formulations.

4.4.1 Types of Models

The interference modeling can be classified as edge-based and node-based, similar to the basic problem. In the edge-based model, the representation of interference is equivalent to a statement such as: "The link *ab* will interfere with the link *cd* if both of them are active on the same channel at the same time". Thus links *ab* and *cd* cannot be scheduled to be active at the same time slot. (Note that this remains true even if *cd* does not interfere with *ab*.) The model is fundamentally symmetric. In the node-based model, the statement of interference constraints take the form: "If node A is transmitting to B and node C is also transmitting, then C will interfere with the reception of B's transmission at A". Note that radio interference actually occurs *at the receiver*, so this model more closely represents the physical reality. However, the model is clearly asymmetric (again, realistic). Naturally, if we choose a node-based formulation for interference, we should correspondingly choose the node-based formulation for the rest of the problem as given in the first formulation of the previous section, and similarly for the edge-based formulations.

The links may be modeled as directed or undirected. If an RTS/CTS/DATA/ACK model of communication is assumed, following several popular wireless LAN protocol models, then each link is constantly being used in both directions, so the links can be naturally considered undirected, and the edge-based formulation is useful. The other alternative is to consider time in finer granularities, and attempt to schedule transmissions from a to b separately from transmissions (including ACKs) from b to a. An edge-based formulation is still possible, but the two directions along each link must be represented by separate variables $l_{n,t}^{(ab)}, l_{n,t}^{(ba)}$. However, the directed link model can also be used with the node-based asymmetric model (unlike the undirected link model).

Finally, we distinguish between the *additive* and *binary* models of interference. So far, we have presented the binary model (usual in literature), that two transmissions either interfere with each other, or do not. However, the physical reality of radio interference is more complex, and the binary model is a simplified approximation. Realistically, while a receiver r is attempting to receive a transmission from a transmitter t, transmission by another node t' which is close to r may be strong enough to garble the original trasnmission; this is adequately represented by the binary model. However, t' may be sufficiently far from r that the effect of its transmission is simply to raise the perceived noise level at r. Whether r successfully recieves the intended transmission or not depends on the SNR at r. However, this SNR depends not only on t', but on *every other node* in the network transmitting at that time, because the noise effects contributed by them add up. This motivates the more realistic *additive model* of interference, as presented in [79].

4.4.2 Asymmetric Models

Symmetric models are available in literature (in particular, above-cited papers), and the interference constraints in such formulations are each of the form:

$$\sum_{ab \in D_i} l_{n,t}^{(ab)} \leq 1, \quad \forall n, t, D_i \in \mathbf{D} \tag{4.10}$$

where each D_i is a set of links each of which will interfere with every other. However, each set of nodes D_i needs to be picked carefully, because the constraint expresses the statement "no more than one of the set D_i of links may be active at any one time". As an example of the possible pitfalls in picking D_i, consider the candidate "the set of links that will interfere with some one link $a'b'$." This generates one D_i for each link $a'b'$, $D(a'b') = \{ab \in A \cup A^I, ab$ interferes with $a'b'\}$. These $\{D_i\}$ are used in the initial formulation in [1], and is implicit in the arguments of [80]. Fig. 4.2 shows an example. In this figure, solid black links

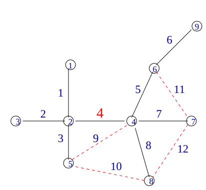


Figure 4.2: An example shows the interference model in [1] is overly restricted

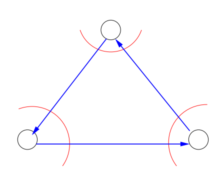


Figure 4.3: Relaxation may be insufficient

are data links and dotted red links are interference links. Suppose there is only a single channel, then it is clear that every other link in the network interferes with the link e_4. (This is because in the protocol model, any two links within two hops of each other are bound to interfere with each other, because there are at least two endpoints that are in transmission or interference range of each other.) Thus the set $D(e_4)$ includes every link in the network, and the corresponding constraint imposes that no more than one edge in the entire network can be active in the same time slot. However, it rules out the possibility that edges e_2 and e_6 can be active at the same time, since they do not interfere with each other. Thus the above definition of $\{D_i\}$ is overly restrictive. (We remark that the simplified formulation of [1], which is used in the rest of the paper, does not exhibit this fallacy.) There is in fact an "if-then" constraint implicit in here, which is correctly expressed as: "*if* link e_4 is active on channel n in timeslot t, *then* none of the links in $D(e_4)$ can be allowed to be active (*else* link e_4 does not constrain the activity of the links in $D(e_4)$, though they may be constrained by other links and each other)".

Motivated by the above discussion, we now propose an asymmetric linear constraint that perfectly represents "if-then" constraints. We show in the next section that this model can be applied to describe the realistic additive physical interference model. This model is given as follows:

$$Ml_{n,t}^{(a'b')} + \sum_{ab \in D(a'b')} l_{n,t}^{(ab)} \leq M, \forall n, t, a'b' \in A, ab \in A \tag{4.11}$$

where $D(a'b')$ is again defined as the set of links that interfere with $a'b'$ (*excluding* $a'b'$ this time). This constraint ensures that *if* $a'b'$ is active in time slot t on channel n, *then* other links that interfere with $a'b'$ can not be active in the same time slot on the same channel. However, if $a'b'$ is not active, this constraint does not constrain the activity of other links in $D(a'b')$ at all, and leaves the other links free to interact governed by other constraints. This is a standard technique to linearize an "if-then" condition. M needs to be only as large as the number of elements of $D(a'b')$ for every constraint; to make it a constant parameter, we can let M be a large number (*e.g.*, $M = |A|$) so that if $l_{n,t}^{(a'b')}$ is 0, the constraint is trivially satisfied. Notice that this constraint does *not* require that $a'b'$ and $ab \in D(a'b')$ *each* interfere with the other, thus it is inherently asymmetric.

4.4.3 Physical Additive Interference Model

We now turn our attention to the more realistic additive interference model. We use the same model as in [79]:

$$\frac{\frac{P_j}{(X(j)-X(i))^\alpha}}{N_a + \sum_{k\neq j} \frac{P_q}{(X(q)-X(i))^\alpha}} \geq \beta \tag{4.12}$$

where P_x is the power of the transmission by node x, $X(x) - X(y)$ is the distance between node x and y, N_a is the noise received by i and β is the SNR threshold. We assume that all radios transmit at a same given power level. Examining the model, it is clear that we need to use the node-based formulation for the overall problem. Further, no symmetric formulation can address this issue, because the success of a node's reception depends on the *simultaneous* behavior of several other nodes. We now translate it into linear constraints using the node formulation as follows, using our asymmetric constraints:

$$\frac{(M-1)P_j \sum_{k,g} l_{k,g,n,t}^{(ji)}}{(X(j)-X(i))^\alpha} + \beta \left(N_a + \sum_{q\neq j} \frac{P_q \sum_k s_{k,n,t}^{(q)}}{(X(q)-X(i))^\alpha} \right)$$
$$\leq \frac{MP_j}{(X(j)-X(i))^\alpha} \forall i, n, t \tag{4.13}$$

Again, M is a large enough number. This constraint ensures that *if* node j is sending data on channel n, *then* the inequality (4.12) must be satisfied. Otherwise, because M is big enough, the constraint is trivially satisfied, and this constraint does not constrain the activity of other nodes.

4.5 Network Optimization With Physical Additive Interference Model

We consider the network design problem that aims at maximizing the concurrent flow, as in [76, 1]. In [1], the authors present a linear relaxation of the constraints and solve the problem using a primal-dual algorithm. Because of the relaxation, the set of constraints becomes a necessary but not sufficient condition. We propose a method that remedies the drawback of relaxation.

Using the notations as above, let the graph be $G(V, E)$. Two nodes are connected by a link if they are within the transmission range of each other. As mentioned in Section 4.4.3, we do not have interference links in this model because of the additive effect of physical interference. We assume the minimum number of radios at two end nodes of a link may be less than the number of channels available (i.e., $\min(K(a), K(b)) \leq u(ab)$, where $u(ab)$ is the number of available channels on link ab. Let T be the scheduling length (number of time slots in each schedule period). γ is a scalar of the traffic demands that needs to be maximized. We present a mathematical formulation of the complete problem (we call it Problem I) as follows.

4.5.1 Problem I - Complete Problem

$$\max \gamma$$

$$\text{s.t.} \sum_b \sum_n r_{ab,n}^{(ij)} - \sum_b \sum_n r_{ba,n}^{(ij)} = \begin{cases} \lambda_{ij} & \text{if } a = i \\ -\lambda_{ij} & \text{if } a = j \\ 0 & \text{otherwise} \end{cases} \quad \forall a, ij \tag{4.14}$$

$$\sum_n r_{ab,n}^{(ij)} \leq \frac{C}{T} \sum_t l_{n,t}^{(ab)} \forall a, t \tag{4.15}$$

$$\sum_n l_{n,t}^{(ab)} \leq u(ab) \forall ab, t \tag{4.16}$$

$$\sum_n \left(\sum_b l_{n,t}^{(ab)} + \sum_b l_{n,t}^{(ab)} \right) \leq k(a) \forall a, t \tag{4.17}$$

$$(M-1)\alpha_{ba} l_{n,t}^{(ba)} + \sum_{qm \in E \setminus ba} \beta \alpha_{qa} l_{n,t}^{(qm)} \leq M\alpha_{ba} - \beta N_0 \forall ba, n, t \tag{4.18}$$

In constraint (4.18), $\alpha_{ab} = \frac{P_a}{(X(a)-X(b))^\alpha}$, which is the power transmitted from node a and received at node b. To prevent a sending node from receiving from other nodes on the same channel at the same time, α_{aa} is P_a, $\forall a$. This also provides us an uniform representation of the interference model. Note that if we assume the schedule length T tends to infinity (or equivalently, in a fixed length of schedule, the length of a time slot is infinitesimal), Equation (4.15) can be changed to

$$\sum_n r_{ab,n}^{(ij)} = \frac{C}{T} \sum_t l_{n,t}^{(ab)} \forall a, t \tag{4.19}$$

$\frac{\sum_t l_{n,t}^{(ab)}}{T}$ is a real number that can be interpreted as the fraction of the schedule length that link ab is active on channel n. It turns out that constraints (4.16), (4.17) and (4.18) can be relaxed by summing both sides over t and substituting $\sum_t l_{n,t}^{(ab)}$ with $\sum_{ij} r_{ab,n}^{(ij)}$ using equation (4.19). Then, we split the problem into two subproblems (Problem II and III).

4.5.2 Problem II - Resource Allocation to Flows Subproblem

$$\max \gamma$$

$$\text{s.t.Equation (4.14) holds} \tag{4.20}$$

$$\sum_n \sum_{ij} \sum_b \left(\frac{r_{ab,n}^{(ij)}}{C} + \frac{r_{ba,n}^{(ij)}}{C} \right) \leq k(a) \forall a \tag{4.21}$$

$$\sum_n \sum_{ij} \frac{r_{ab,n}^{(ij)}}{C} \leq u(ab) \forall ab \tag{4.22}$$

$$(M-1)\alpha_{ba} \sum_{ij} r_{ba,n}^{(ij)} + \sum_{qm \in E \backslash ba} \beta \alpha_{qa} \sum_{ij} r_{qm,n}^{(ij)} \leq C \left(M\alpha_{ba} - \beta N_0 \right) \forall ba, n \tag{4.23}$$

The feasible region of problem II is the intersection of three feasible regions, the one described by the flow balance constraints (4.14), the one described by number of radio and number of channels constraints (4.21) and (4.22), and the one described by the physical interference constraint (4.23). It is interesting to observe that $\sum_n r_{ab,n}^{(ij)}/C$ is in fact a linear relaxation of $l_{n,t}^{(ab)}$. Thus, a feasible solution to problem II may be unschedulable. Fig. 4.3 illustrates an example (we assume interferences do not exist, to study only the effect of the number of radios and channels constraints). There are three nodes in the figure, each node is equipped with one radio, each link has one channel. Obviously, it satisfies the relaxed constraints if each link is active half of the schedule length. However, because of Constraint (4.17), this is unschedulable. Therefore, it motivates us to add additional constraints such that the feasible region is integral. On the other hand, because the constraints imposed on physical interference have fractional coefficients, this motivates us to relax these constraints.

Leaving, for the moment, the physical interference constraints out of consideration, it is easy to see that this is a capacitated b-matching problem. Therefore, by adding *blossom inequalities* [81], we can obtain a set of necessary and sufficient conditions for the problem without interference. In other words, by solving the linear relaxation problem, we can obtain the fraction of the schedule length that the channel is active on a specific channel such that the constraints imposed by the number of radios at each node and the number of channels on each link are obeyed. In that case, we shall have the following guarantee:

Theorem 4.5.1. *Suppose the time any given channel is active on a link, expressed as a fraction of the schedule length, is a, and the length of a timeslot similarly represented is b, in a solution obtained as above. If a is an integer multiple of b, there exists a schedule corresponding to this solution such that the Constraints (4.17) and (4.16) are obeyed in every time slot.*

Proof. Since the polytope described by constraints (4.21), (4.22) and blossom inequalities is integral (vertices are 0-1), a feasible solution is a convex combination of the vertices [82]. That is:

$$s = \sum_i a_i v_i, \text{ and } \sum_i a_i = 1, a_i \geq 0 \forall i,$$

where v_i is a vertex. Let m be the number of slots in a schedule length, according to the assumption, $a_i m$ is an integer. Then, a matrix is formed by expanding row i of the basis matrix to $a_i m$ copies, for all i. Let the transpose of the matrix be $X = [x_{ij}]$, where $x_{ij} = 1$ (respectively, $x_{ij} = 0$) means the link-channel pair i is active (respectively, inactive) at time slot j. Then, X is a feasible schedule because of the following reasons. Firstly, each column is a 0-1 vertex, therefore both (4.21) and (4.22) are obeyed. Secondly, $XE = s'$, where E is an m-vector with every element of the vector equal to $1/m$. Therefore, the fraction of the time that a link-channel pair is active is satisfied. ■

The blossom inequalities are given as follows:

$$\begin{aligned}\sum_{ab\in E(W)}\sum_{n}\sum_{ij} r_{ab,n}^{(ij)} + \sum_{ab\in\delta(W)}\sum_{n}\sum_{ij} r_{ab,n}^{(ij)} \leq \\ C\left\lfloor\frac{\sum_{a\in W}k(a)+\sum_{ab\in\delta(W)}u(ab)}{2}\right\rfloor, \\ \forall W\subset V, \text{with} \sum_{a\in W}K(a)+\sum_{ab\in\delta(W)}u(ab) \text{ odd,}\end{aligned} \tag{4.24}$$

where $E(W)$ (respectively, $\delta(W)$) is the set of edges with both end-vertices (respectively, exactly one end-vertex) in W. However, the number of the blossom inequalities can be very large. To circumvent this difficulty, we use the separation algorithm developed by Letchford et al [81]. First, we dualize Problem II. Let

$$\begin{aligned}\mathcal{L}_{ab,n} = & \frac{1}{C}\left(\frac{\sum_{W:ab\in E(W)}s_W+\sum_{W:ab\in\delta(W)}s_W}{\left\lfloor\left(\sum_{a\in W}k(a)+\sum_{ab\in\delta(W)}u(ab)\right)/2\right\rfloor}+\right. \\ & \frac{y_a}{k(a)}+\frac{y_b}{k(b)}+\frac{z_{ab}}{u(ab)}+\frac{(M-1)\alpha_{ab}p_{ab,n}}{M\alpha_{ab}-\beta N_o}+ \\ & \left.\sum_{qm\in E\backslash ab}\frac{\beta\alpha_{am}p_{qm,n}}{M\alpha_{qm}-\beta N_o}\right)\end{aligned} \tag{4.25}$$

where, $x_a^{(ij)}, y_a, z_{ab}, p_{ab,n}, s_W$ are dual variables to constraints (4.14), (4.21), (4.22), (4.23) and (4.24) respectively. We obtain the dual problem:

$$\min \sum_a y_a + \sum_{ab} z_{ab} + \sum_{ab}\sum_n p_{ab,n} + \sum_W s_W$$

$$\text{s.t.} \sum_{ij}\left(x_i^{(ij)} - x_j^{(ij)}\right)\lambda_{ij} \geq 1 \tag{4.26}$$

$$\mathcal{L}_{ab,n} \geq x_b^{(ij)} - x_a^{(ij)} \tag{4.27}$$

$$y_a, z_{ab}, p_{ab,n}, s_W \geq 0 \forall a, ab, n, W$$

It is also interesting to note that, in directed graphs, the blossom inequalities may not exist. However, in our problem, the number of channels on each link and the number of radios on each node are shared by connections of both directions (because the interference model prevents two connections from taking the same channel on the same link), therefore we can simply treat the graph as undirected. The directions of flows will be taken care of by the flow conservation constraints and interference constraints.

Next we describe the algorithm. It merely consists of repeatedly locating violated blossom ineqalities using the efficient algorithm from [81], and adding upto a fixed number of them.

Table 4.2: Algorithm that solves Problem II

```
Loop: Solve Problem II by a primal-dual algorithm
      If a certain number of loops is reached, break
      Run the separation algorithm
      If a violated blossom inequality is found
           add the inequality to the problem, goto Loop
      else break
end loop
```

For the rest of this problem, we simply adopt an existing good approach, which we briefly describe below. The primal-dual algorithm is a fully polynomial time approximation algorithm (FPTAS) that is used in [1]. The analysis follows the work of Garg and Könemann [83]. For the sake of brevity, we first introduce the notation. Let the set of constraints be denoted as J. $f(ab, n)$ is the flow on the channel n over link ab. Let l be a length function. Let $j \in J$ denote a constraint. Then, constraint j can be represented as a generalized form

$$A_j R \leq RHS_j, \tag{4.28}$$

where vector $R = \{\sum_{ij} r_{ab,n}^{(ij)}\}$, and RHS_j is simply the appropriate Right Hand Side of the constraint j. Let R_j be the set of link-channel pairs on which the constraint j is imposed. A path P is formed by a set of link-channel pairs. The maximum amount of flow allowed on path P with imposed constraint j, $F(P, j)$, is determined by $RHS_j(A_j\mathbf{1})^{-1}$, if $P \cap R_j \neq \emptyset$, where $\mathbf{1}$ is an index vector whose element is 1 if the corresponding link-channel pair is in $P \cup R_j$. Let δ be a predetermined constant. The algorithm is described in table 4.3.

The separation algorithm is described in table 4.4. Given the amount of traffic flow on each link obtained by the primal-dual algorithm, we find a set of nodes such that Constraint (4.24) is most violated. This is done by forming a cut-tree for the graph with an additional dummy node. Then, the blossom inequality condition is checked on the cut induced by each edge of the cut-tree. For details, the reader is referred to [81]. Because the

Table 4.3: Primal-Dual Algorithm

Initialization: $l(j) = \delta \forall j$
While $\sum_j l(j) \leq 1$
For $ij : \lambda_{ij} > 0$
$\lambda = \lambda_{ij}$
While $\lambda > 0$
Set weights of $\mathcal{L}_{ab,n}$ on each link-channel pair (ab,n)
Compute P^*, the shortest path from i to j
Let $u = F(P^*, j)$
$\delta = \min\{\lambda, u\}$
$\lambda \leftarrow \lambda - \delta$
$f(ab, n) \leftarrow f(ab, n) + \delta \forall (ab, n) \in P^*$
$l(j) \leftarrow l(j) \left(1 + \frac{\delta}{F(P^*,j)}\right) \forall j : P^* \cap R_j \neq \emptyset$
End While
End for
$t \leftarrow t + 1$
End While
Compute $\rho = \max_j \sum_{(ab,n) \in R_j f(ab,n)/C}$
Output $\lambda^* = t/\rho$

number of blossom inequalities can be large, a maximum number of iterations is introduced to force the end of the algorithm. However, we see in Section 4.6 that introducing only a reasonable number of blossom inequalities usually helps significantly.

4.5.3 Problem III - Scheduling Subproblem

$$\max \sum_{ab} \sum_{n} \sum_{t} l_{n,t}^{(ab)}$$

$$\text{s.t.} \frac{\sum_{ij} r_{ab,n}^{(ij)}}{C} \geq \frac{\sum_t l_{n,t}^{(ab)}}{T} \forall ab, n \tag{4.29}$$

$$(M-1)\alpha_{ba} l_{n,t}^{(ba)} + \sum_{qm \in E \backslash ba} \beta \alpha_{qa} l_{n,t}^{(qm)}$$

$$\leq M\alpha_{ba} - \beta N_0 \forall a, n, t \tag{4.30}$$

Problem III aims at solving the scheduling problem. Note that in solving Problem II, we assumed that T tends to infinity and relaxed the integer requirements. The optimization gap has been reduced due to the introduction of blossom inequalities, however it is hard to

Table 4.4: Separation algorithm for capacitated b-matching problem

Input: Graph $G(V,E)$, $u(ab)\forall ab$, $k(a)\forall a$, and $r^{(ij)}_{ab,n}\forall ij, ab, n$,
Output:
Let $r^*_{ab} = \sum_n \sum_{ij} r^{(ij)}_{ab,n}$, $w_{ab} = \min\{r^*_{ab}, u(ab) - r^*_{ab}\}$
Construct graph $G^+(V^+, E^+)$ by adding a dummy node v to G,
$\forall a \in V$, add an edge av with weight $k(a) - \sum_n \sum_{ij} \sum_b r^{(ij)}_{ab,n}$
Compute a *cut-tree* for G^+ with *terminal nodes* V^+ using *Gomory & Hu's algorithm*
For each edge of the *cut-tree*, let the induced cut in G be $\delta(W)$, do
Check $(\delta(W))$
If exists a violated blossom inequality using $\delta(W)$,
Output the inequality, STOP,
End if
End for

find additional constraints for the physical interference constraints. Since a feasible solution of Problem II may be unschedulable, in Problem III, given the bound provided by solving Problem II, a real schedule is found such that the objective function value is as close to the bound as possible.

We solve the problem by a *Lagrangian Relaxation* method. We define the Lagrangian subproblem $\mathcal{L}(\mu)$ as follows:

$$\max \sum_{ab,n,t} \left(1 + \mu_{ab}(M-1)\alpha_{ab} + \sum_{i\neq b} \mu_{ai}\beta\alpha_{ai}\right) l^{(ab)}_{n,t}$$

$$\text{s.t.} \quad \frac{\sum_t l^{(ab)}_{n,t}}{T} \leq \frac{\sum_{ij} r^{(ij)}_{ab,n}}{C} \tag{4.31}$$

Then, the objective is

$$\min_{\mu\geq 0} \mathcal{L}(\mu) \tag{4.32}$$

If T is fixed, Constraint (4.31) is not integral. That is, we may be unable to find an optimal integer solution for the Lagrangian relaxed subproblem because the RHS of Constraint (4.31) may be fractional. Thus, Constraint (4.31) is modified as:

$$\sum_t l^{(ab)}_{n,t} \leq \left\lfloor T \frac{\sum_{ij} r^{(ij)}_{ab,n}}{C} \right\rfloor \tag{4.33}$$

If T is infinite, i.e., in a fixed schedule length, the length of a time slot can be infinitesimal, we can set T be the smallest number such that the RHS is integral. That is, we can pick the shortest schedule which will precisely maintain the link allocations as computed. The final result is post-processed if necessary to make it feasible as follows. For each $l_{n,t}^{(ab)}$, check if constraint (4.18) is violated or not. If it is violated, find the link $qm \in A \setminus ab, l_{n,t}^{(qm)} = 1$ such that $\alpha_{qa} l_{n,t}^{(qm)}$ is maximal. Set $l_{n,t}^{(qm)}$ to 0 and reduce all traffic demands ij traversing qm (i.e., $r_{ab,n}^{(ij)} > 0$) by δ such that constraint (4.15) is satisfied.

4.6 Numerical Results

In this section, we present the numerical results. We generate random grid networks. The number of radios at each node and the number of channels on each link are randomly generated. We test our algorithm with various parameters. Specifically, the SNR varies from 4dB to 10dB. The powers sent by all nodes are assumed to be the same, which is varied between 50mW to 100mW. The ambient noise is assumed to be 0mW, that is we isolate the effect of the network nodes only. The traffic matrix is also randomly generated.

First, we evaluate the performance of our algorithm. We obtain a lower bound of the solution by solving the original problem using CPLEX mixed integer programming solver. An upper bound is achieved by solving the linear version of the problem without adding blossom inequalities. The upper bound and the solution given by our algorithm are normalized to the lower bound and displayed in Fig. 4.4. Because of the computation complexity, the comparison is performed on a grid network that consists of 5×5 nodes. As expected, our solution falls in the gap of the bounds. In addition, we observe that the gap between the bounds can be large. In Fig. 4.5, we compute the maximum concurrent flow

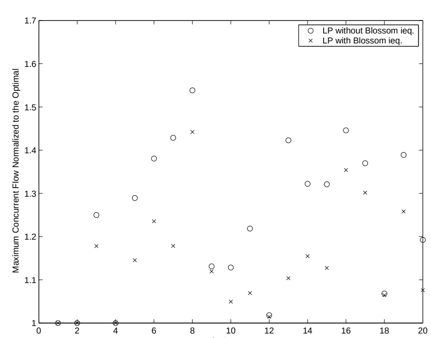


Figure 4.4: Bounds On the Maximum Concurrent Flow

with various number of radios and channels using our algorithm. The network tested is a grid with same number of radios at each node and same number of channels on each link. The number of radios and the number of channels are varied from 1 to 5. The result shows that the number of radios is more constrained. It is because the degree of each node in a grid is at least 2.

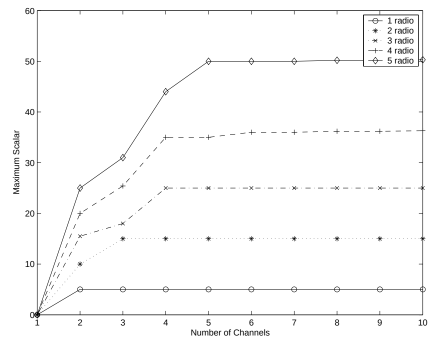


Figure 4.5: Maximum concurrent flow vs. number of radios vs. number of channels

In Fig 4.6, we compute the maximum concurrent flow with different selections of SNR threshold and power decay. We keep the distance of each edge in the grid network unchanged while changing α, the power decay factor (*i.e.*, power decays with r as $\frac{1}{r^\alpha}$). It shows that as α increases, the maximum objective value also increases, due to the decrease of interference. However, when α is too large, the network becomes unconnected. The figure also shows that when SNR threshold is larger, the maximum objective value is smaller. This is obvious because a larger SNR threshold implies the node is more sensitive to interference.

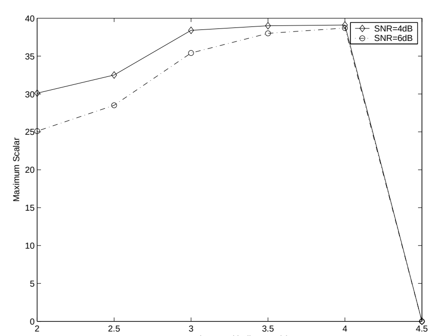


Figure 4.6: Maximum concurrent flow vs. power decay vs. SNR threshold

The following figure shows how the number of blossom inequalities affects the optimal objective value. We generate grid networks with 10×10 nodes, and use different number of iterations (that is, the maximum number of blossom inequalities that can be added) to solve each instance. The result shows that when the number of iterations is increased, the optimal objective value is also increased (in most cases). Our experiments show that in every instance the algorithm stops when the maximum number of iterations is reached. It suggests that, if more constraints are added, the optimal objective value of Problem II can be further optimized. However, the incremental benefit is seen to be small. Further, because of the heuristic nature of the Lagrangian relaxation method used to solve the Problem III, the effort taken in solving the previous subproblem may not be worthwhile. Thus we conclude that a reasonable number of blossom inequalities suffices for our approach.

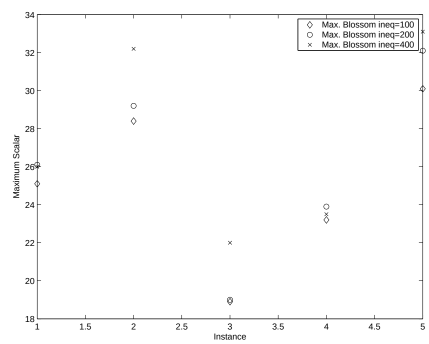


Figure 4.7: Maximum concurrent flow vs. number of blossom inequalities

Part IV

Conclusion

The network design problem is highly complicated and often formulated as some kind of optimization problems with various objectives. It takes constriants such as the network status, resource availability, traffic models as the input. These constraints are general problem and network specific. As opposed to online algorithms that are often heuristics, the network design problem is usually solved by some highly complicated models and time consuming algorithms. In our study, we conducted research on both optical and wireless networks.

In optical networks, we studied the dynamic traffic grooming problem, which is becoming a hot research topic in the community. The dynamic traffic grooming problem is an important area to the research community as well as to service providers. In today's WDM networks, the increasing number of wavelengths available on an optical fiber and various optical/electronic equipment with different functionalities enable networks that are not only increasingly complex but also more and more agile. Accordingly, they provide more opportunity to balance the complexity (usually translated into cost) and the agility. In this sense, the dynamic traffic grooming problem is envisioned to be an essential area in the future. In this dissertation, we have presented a literature survey of the dynamic traffic grooming area. We started from the physical layer by discussing different optical equipment and their architectures. Then we classified the dynamic traffic grooming problem into the design and analysis problems, and discuss the formulation of the design problem as optimization or decision problems. Following the classification, we surveyed the literature throughly.

Although the dynamic traffic grooming problem has already been extensively studied, many practically important problems worthy of study still remain open. In the analysis class, models with limited complexity are needed. Models of high complexity are theoretically useful but may not see extensive practical application. In a similar sense, the models need to take the mesh topology, multi-rate traffic model and link load correlation into consideration. Current approaches often make restrictive assumptions such as very simple topologies, or link independence, that make them less practically useful to the network designer, even though they may provide good insight into the nature of the problem. In addition, since networks generally are upgraded instead of built from scratch, we expect that the network may very often have a heterogeneous architecture. Because of the distinctions between traffic grooming networks and traditional data/circuit networks, this problem is of particular interest.

In the class of design problems, we believe that under the umbrella of the dynamic traffic grooming problem, many more interesting and practical problems remain to be discovered and solved. For example, some particular traffic models may be of practical interest. As we mentioned above, the Scheduled Lightpath Demand (SLD) traffic model has been generalized in several directions, including that of subwavelength traffic. However, some interesting and practically important generalizations (such as sliding window scheduled demands) remain unaddressed in the subwavelength context. Another interesting problem is that of translating QoS requirements from different levels in the network. It is envisioned that GMPLS will be widely deployed as a management layer in next generation networks. Therefore, approaches to dynamically groom subwavelength LSPs onto lightpaths while taking the QoS requirements (*e.g.*, delay) into consideration needs to be studied.

As the field evolves, traffic grooming may be seen as a general problem of network design where the cost component is largely concentrated into specialized network node equipment (as opposed to bandwidth, in yesteryear's networks). In the near future, minimizing OEO may well cease to be a worthwhile goal, if device technology makes appropriate advances. However, the presence of a large amount of dark fiber in the ground makes it likely that some other nodal equipment, such as optical drop-and-continue, wavelength converters, OTDM switches, or some other emerging technology will dominate network costs.

The issue of multicast and broadcast grooming has been addressed in the static context and there are some works in the dynamic context as well. However, it continues to be comparatively less explored. The increasing trend of using the Internet for content distribution for media applications requiring SLAs may well cause interest in this area to grow.

Another interesting development is likely to come from waveband grooming; wavebands or coarse wavelengths are optical channels created by less selective optical filters and transponder equipments, so that a number of usual lightpaths can be optically forwarded with the use of a single such waveband port. Thus the waveband introduced yet a third layer of topology in the design problem, and waveband grooming has already drawn the attention of researchers in the static context. Literature is soon likely to appear on dynamic waveband grooming.

Lastly, the lessons learned from traffic grooming may be applied to other areas of research in future. The emergence of wireless networks as viable metro area networks makes such wireless networks, and heterogeneous networks formed of optical and wireless domains, an interesting area of research. Such an environment is typically more dynamic than wireline networks. The desire to provide SLAs to wireless LAN customers introduces the theme of QoS to sub-circuit flows, which is a distinguishing characteristic of traffic grooming. In short, we expect many interesting and far-reaching research results to develop out of the comparatively new research area of dynamic traffic grooming. We hope that our survey, in a modest way, will help researchers newly entering this field.

We have also investigated a specific problem of spare capacity allocation in support of dynamic traffic grooming in optical networks. We have formulated the problem as a mathematical programming problem, and shown that it is a separable convex problem for the single logical link case, and a simple algorithm to obtain the optimal integer solution exists. We have shown that this tractabilty is lost for a general logical topology, and have provided good practical heuristics for that case. Numerical results show that our approaches produce good results.

There are many directions for future work in this area, such as finding other algorithms for the general topology case, addressing the simplifying assumptions we have made such as state independence, etc.; and we are following some of them. We hope to report more results in this area soon.

In wireless networks, one of the major difficulties comes from the MAC protocol. Comparing to their optical counterpart, wireless devices are much less expensive, therefore can be deployed in large scale. In addition, the design of the WMNs has to take the environment into consideration because wireless devices are more subject to these conditions.To satisfy the need of large scale WMNs, we have considered the WMN design problem under the physical additive interference model, which has so far not been addressed in WMN design. We have presented a new asymmetric model, and an algorithm based on viewing part of the problem can as a generalized matching problem. The performance of our algorithm was validated by numerical experimentations.

Bibliography

[1] M. Kodialam and T. Nandagopal, “Characterizing the capacity region in multi-radio multi-channel wireless mesh networks,” in *Proc. of MobiCom ’05*, 2005.

[2] R. Ramaswami and K. Sivarajan, *Optical networks: a practical perspective.* Morgan Kaufmann Publishers, 1998.

[3] B. Mukherjee, *Optical communication networks.* McGraw-Hill, 1997.

[4] R. Dutta and G. Rouskas, “Traffic grooming in WDM networks: Past and future,” *IEEE Network*, pp. 46–56, 2002.

[5] L. Berger (Ed.), “Generalized multi-protocol label switching (GMPLS) signaling functional description.” Internet Enginerring Task Force RFC, January 2003, no. 3471.

[6] E. Mannie (Ed.), “Generalized multi-protocol label switching (GMPLS) architecture.” Internet Enginerring Task Force RFC, October 2004, no. 3945.

[7] E. Mannie and D. Papadimitriou, “Generalized multi-protocol label switching (GMPLS) extensions for synchronous optical network (SONET) and synchronous digital hierarchy (SDH) control.” Internet Enginerring Task Force RFC, October 2004, no. 3946.

[8] D. Papadimitriou, J. Drake, J. Ash, A. Farrel, and L. Ong, “Requirements for generalized mpls (GMPLS) signaling usage and extensions for automatically switched optical network (ASON).” Internet Enginerring Task Force RFC, July 2005, no. 4139.

[9] K. Kompella and Y. Rekhter (Eds.), “Routing extensions in support of generalized multi-protocol label switching (GMPLS).” Internet Enginerring Task Force RFC, October 2005, no. 4202.

[10] K. Zhu, H. Zang, and B. Mukherjee, "A comprehensive study on next-generation optical grooming switches," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 7, pp. 1173 – 86, 2003.

[11] W. Goralski, *SONET*. Mc-Graw Hill, 2000.

[12] R. Srinivasan and A. Somani, "Request-specific routing in wdm grooming networks," *2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No.02CH37333)*, vol. 5, pp. 2876 – 80, 2002.

[13] B. Bacque and D. Oprea, "R-OADM architecture: Now you can control the light," 2003, architectural white paper, available at http://www.tropicnetworks.com/library/pdf/ROADM_White_Paper_May_03.pdf.

[14] A. Rasala and G. Wilfong, "Strictly non-blocking WDM cross-connects for heterogeneous networks," *Proceedings of the Thirty Second Annual ACM Symposium on Theory of Computing*, pp. 514 – 23, 2000.

[15] K. Zhu, H. Zhu, and B. Mukherjee, "Traffic engineering in multigranularity heterogeneous optical wdm mesh networks through dynamic traffic grooming," *IEEE Network*, vol. 17, no. 2, pp. 8 – 15, 2003.

[16] R. Srinivasan and A. Somani, "A generalized framework for analyzing time-space switched optical networks," *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 1, pp. 179 – 88, 2001.

[17] S. Ramasubramanian and A. Somani, "Analysis of optical networks with heterogeneous grooming architectures," *IEEE/ACM Transactions on Networking*, vol. 12, no. 5, pp. 931 – 43, 2004.

[18] H. Madhyastha and C. Siva Ram Murthy, "Efficient dynamic traffic grooming in service-differentiated wdm mesh networks," *Computer Networks*, vol. 45, no. 2, pp. 221 – 35, 2004.

[19] J. Kuri, N. Puech, and M. Gagnaire, "Routing and grooming of scheduled lighpath demands in a multi-granularity switching network: a mathematical model," *Optical Network Design and Modeling*, 2005.

[20] X. Huang, F. Farahmand, and J. Jue, "An algorithm for traffic grooming in wdm mesh networks with dynamically changing light-trees," *GLOBECOM '04. IEEE Global Telecommunications Conference (IEEE Cat. No.04CH37615)*, vol. 3, pp. 1813 – 17, 2004.

[21] A. Khalil, C. Assi, A. Hadjiantonis, G. Ellinas, N. Abdellatif, and M. Ali, "Multicast traffic grooming in WDM networks," *Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513)*, vol. 2, pp. 785 – 8, 2004.

[22] L. Zhang and G.-S. Poo, "A dynamic traffic grooming algorithm in multigranularity heterogeneous optical wdm mesh networks," *ICICS-PCM 2003. Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing and Fourth Pacific-Rim Conference on Multimedia (IEEE Cat. No.03EX758)*, vol. 2, pp. 1286 – 9, 2003.

[23] H. Kim, S. Ahn, and J. Chung, "Dynamic traffic grooming and load balancing for gmpls-centric all optical networks," *Knowledge-Based Intelligent Information and Engineering Systems. 8th International Conference, KES 2004. Proceedings (Lecture Notes in Artificial Intelligence Vol.3215)*, vol. 3, pp. 38 – 44, 2004.

[24] S. Huang and R. Dutta, "Research problems in dynamic traffic grooming in optical networks," in *Proceedings of First International Workshop on Traffic Grooming,(BROADNETS'04)*, 2004.

[25] J.-Q. Hu, "Traffic grooming in WDM ring networks: A linear programming solution," *Journal of Optical Networks*, vol. 1, 2002.

[26] R. Berry and E. Modiano, "Reducing electronic multiplexing costs in SONET/WDM rings with dynamically changing traffic," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1961 – 71, 2000.

[27] N. Srinivas and C. Siva Ram Murthy, "Design and dimensioning of a WDM mesh network to groom dynamically varying traffic," *Photonic Network Communications*, vol. 7, no. 2, pp. 179 – 91, 2004.

[28] B. Wang, T. Li, X. Luo, and Y. Fan, "Traffic grooming under a sliding scheduled traffic model in WDM optical networks," in *Opticomm*, Oct 2004.

[29] G. Sasaki and O. Gerstel, "Minimal cost WDM sonet rings that guarantee no blocking," in *Optical Networks Magazine*, vol. 4, Oct 2000.

[30] S. Thiagarajan and A. Somani, "Capacity fairness of WDM networks with grooming capabilities," *Optical Networks Magazine*, vol. 2, no. 3, pp. 24 – 32, 2001.

[31] Y. Xu, S.-C. Xu, and B.-X. Wu, "Strictly nonblocking grooming of dynamic traffic in unidirectional SONET/WDM rings using genetic algorithms," *Computer Networks*, vol. 41, no. 2, pp. 227 – 45, 2003.

[32] S. Huang and R. Dutta, "Dynamic traffic grooming - a survey," *NCSU CSC department Technical Report*, no. TR-2006-15, 2006, available from *http://www.csc.ncsu.edu/research/tech-reports/README.html*.

[33] K. Mosharaf, J. Talim, and I. Lambadaris, "A markov decision process model for dynamic wavelength allocation in all-optical WDM networks," in *GLOBECOM2003*, vol. 5, Dec 2003.

[34] A. Washington and H. Perros, "Call blocking probabilities in a traffic groomed tandem optical network," *Special issue dedicated to the memory of Professor Olga Casals, Blondia and Stavrakakis (Eds.) Journal of Computer Networks*, vol. 45, 2004.

[35] C.Xin, C.Qiao, and S.Dixit, "Analysis of single-hop traffic grooming in mesh WDM optical networks," in *Opticomm*, Oct 2003.

[36] C.Xin and C.Qiao, "Performance analysis of multi-hop traffic grooming in mesh WDM optical networks," in *The 12th International Conference on Computer Communications and Networks*, 2003.

[37] R. Srinivasan and A. Somani, "Analysis of multi-rate traffic in WDM grooming networks," *Proceedings Eleventh International Conference on Computer Communications and Networks (Cat. No.02EX594)*, pp. 296 – 301, 2002.

[38] C. Xin, B. Wang, X. Cao, and J. Li, "Formulation of multi-hop dynamic traffic grooming in WDM optical networks," *Proceedings of the Second International IEEE/Create-Net Workshop on Traffic Grooming*, 2005.

[39] W. Yao, G. Sahin, M. Li, and B. Rarmamurthy, "Analysis of multi-hop traffic grooming in WDM mesh networks," *Proceedings of BroadNets, the Second International IEEE/Create-Net Conference on Broadband Networks*, 2005.

[40] S. Thiagarajan and A. Somani, "A capacity correlation model for WDM networks with constrained grooming capabilities," *ICC 2001*, vol. 5, pp. 1592–1596, 2001.

[41] K. Mosharaf, J. Talim, and I. Lambadaris, "A call admission control for service differentiation and fairness management in WDM grooming networks," *Proceedings. First International Conference on Broadband Networks*, pp. 162 – 9, 2004.

[42] G. Sasaki and T. Lin, "A minimal cost WDM network for incremental traffic," in *Information Theory and Communications Workshop,Proceedings of the 1999 IEEE*, June 1999.

[43] R. Cigno, E.Salvadori, and Z.Zsoka, "Elastic traffic effects on WDM grooming algorithms," in *Globecom*, 2004.

[44] R. Kandula and G. Sasaki, "Grooming of dynamic tributary traffic in WDM rings with rearrangements," Presented at the 39th Annual Allerton Confrerence on Communication, Control, and Computing, Monitcello IL, Oct 2001.

[45] W. Yao and B. Ramamurthy, "Rerouting schemes for dynamic traffic grooming in optical WDM mesh networks," *GLOBECOM '04. IEEE Global Telecommunications Conference (IEEE Cat. No.04CH37615)*, vol. 3, pp. 1793 – 7, 2004.

[46] H. Wen, R. He, L. Li, and S. Wang, "Dynamic traffic-grooming algorithms in wavelength-division-multiplexing mesh networks," *Journal of Optical networking*, vol. 2, no. 4, 2003.

[47] K. Elsayed, "Dynamic routing, wavelength, and fibre selection algorithms for multifibre WDM grooming networks," *IEE Proceedings-Communications*, vol. 152, no. 1, pp. 119 – 27, 2005.

[48] N. Bhide, K. Sivalingam, and T. Fabry-Aztalos, "Routing mechanisms employing adaptive weight functions for shortest path routing in optical WDM netowrks," *Photonic Networks Communications*, vol. 3, pp. 227–236, 2001.

[49] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "Dynamic traffic grooming in WDM mesh networks using a novel graph model," *Optical Networks Magazine*, vol. 4, no. 3, pp. 65 – 75, 2003.

[50] F. Farahmand, X. Huang, and J. Jue, "Efficient online traffic grooming algorithms in WDM mesh networks with drop-and-continue node architecture," *Proceedings. First International Conference on Broadband Networks*, pp. 180 – 9, 2004.

[51] B. Chen, W.-D. Zhong, and S. Bose, "A path inflation control strategy for dynamic traffic grooming in IP/MPLS over WDM network," *IEEE Communications Letters*, vol. 8, no. 11, pp. 680 – 2, 2004.

[52] R. He, H. Wen, and L. Li, "Fairness-based dynamic traffic grooming in WDM mesh networks," *APCC/MDMC '04. The 2004 Joint Conference of the 10th Asia-Pacific Conference on Communications and the 5th International Symposium on Multi-Dimensional Mobile Communications Proceeding*, vol. 2, pp. 602 – 6, 2004.

[53] B. Chen, S. Bose, and W.-D. Zhong, "Priority enabled dynamic traffic grooming," *IEEE Communications Letters*, vol. 9, no. 4, pp. 366 – 8, 2005.

[54] I. Cerutti, A. Fumagalli, and S. Sheth, "Performance versus cost analysis of WDM networks with dynamic traffic grooming capabilities," *Proceedings. 13th International Conference on Computer Communications and Networks (IEEE Cat. No.04EX969)*, pp. 425 – 30, 2004.

[55] R. Srinivasan and A. Somani, "Dynamic routing in WDM grooming networks," *Photonic Network Communications*, vol. 5, no. 2, pp. 123 – 35, 2003.

[56] W. Yao and B. Ramatnurthy, "Constrained dynamic traffic grooming in WDM mesh networks with link bundled auxiliary graph model," *2004 Workshop on High Performance Switching and Routing (IEEE Cat. No.04TH8735)*, pp. 287 – 91, 2004.

[57] K. hong Liu and Y. Xu, "A new approach to improving the grooming performance with dynamic traffic in sonet rings," *Computer Networks*, vol. 46, no. 2, pp. 181 – 95, 2004.

[58] T.-T. N. Thi, T. T. Minh, Q.-D. Ho, and M.-S. Lee, "A time and cost efficient dynamic traffic grooming algorithm for optical mesh networks," *Proceedings of the Second International IEEE/Create-Net Workshop on Traffic Grooming*, 2005.

[59] Q.-D. Ho and M.-S. Lee, "Practical dynamic traffic grooming in large WDM mesh networks," *Proceedings of the Second International IEEE/Create-Net Workshop on Traffic Grooming*, 2005.

[60] O. Gerstel, R. Ramaswami, and G. Sasaki, "Cost-effective traffic grooming in WDM rings," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 618 – 30, 2000.

[61] S. Zhang and B. Ramamurthy, "Dynamic traffic grooming algorithms for reconfigurable SONET over WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 7, pp. 1165 – 72, 2003.

[62] A. Gencata and B. Mukherjee, "Virtual-topology adaptation for WDM mesh networks under dynamic traffic," *IEEE/ACM Transactions on Networking*, 2003.

[63] E. A. Doumith, M. Gagnaire, O. Audouin, and R. Douville, "Network nodes dimensioning assuming electrical traffic grooming in an hybrid OXC/EXC WDM network," *Proceedings of the Second International IEEE/Create-Net Workshop on Traffic Grooming*, pp. 177–187, 2005.

[64] M. El Houmaidi, M. Bassiouni, and G. Li, "Optimal traffic grooming in WDM mesh networks under dynamic traffic," *Optical Fiber Communication Conference (OFC) (IEEE Cat. No.04CH37532)*, vol. 2, 2004.

[65] I. Widjaja, I. Saniee, L. Qian, A. Elwalid, J. Ellson, and L. Cheng, "A new approach for automatic grooming of SONET circuits to optical express links," *2003 IEEE International Conference on Communications (Cat. No.03CH37441)*, vol. 2, pp. 1407 – 11, 2003.

[66] A. Chiu and E. Modiano, "Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks," *IEEE Journal on Lightwave Technology*, 2000.

[67] R. Dutta, , S. Huang, and G. Rouskas, "Optimal traffic grooming in elemental network topologies," *Opticomm*, pp. 46–56, 2003.

[68] R. Sabella, P. Iovanna, G. Oriolo, and P. D'Aprile, "Strategy for dynamic routing and grooming of data flows into lightpaths in new generation network based on the GMPLS paradigm," *Photonic Network Communications*, vol. 7, no. 2, pp. 131 – 44, 2004.

[69] C.Zhao and J.Q.Hu, "Traffic grooming for WDM rings with dynamic traffic," manuscript, 2003.

[70] J. Keilson, *Markov Chain Models - Rarity and Exponentiality.* Springer-Verlag, 1979.

[71] *MATLAB User Manual.* Simulink Corp.

[72] M. Kodialam and T. Nandagopal, "Characterizing the achievable rates in multi-hop wireless networks: The joint routing and scheduling problem," in *Proc. of MobiCom '03*, 2003.

[73] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Mobicom '03*, 2003.

[74] A. Raniwala and T. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proc. of IEEE Infocom '05*, 2005.

[75] A. Raniwala, K. Gopalan, and T. cker Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," in *Mobile Computing and Communications Review*, vol. 8, 2005.

[76] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in mulit-radio wireless mesh networks," in *Proc. of MobiCom '05*, 2005.

[77] P. Kyasanur and N. H.VAidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," in *IEEE WCNC '05*, 2005.

[78] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio multi-channel wireless mesh networks," in *MobiCom '04*, 2004.

[79] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, 2000.

[80] J. Jun and M. L. Sichitiu, “The nominal capacity of wireless mesh networks,” *IEEE Wireless Communications Magazine, Special Issue on: Merging IP and Wireless Networks*, pp. 8–14, Oct 2003.

[81] A. N. Letchford, G. Reinelt, and D. O. Theis, “A faster exact separation algorithm for blossom inequalities,” in *LECTURE NOTES IN COMPUTER SCIENCE*, 2004.

[82] E. J., “Maximum matching and a polyhedron with 0-1 vertices,” in *J. Res. Nat. Bur. Standards*, 1965, pp. 125–130.

[83] N. Garg and J. Könemann, “Faster and simpler algorithms for multi-commodity flow and other fractional packing problems,” in *Proc. of 39th Annual Symposium on Foundations of Computer Science*, 1998.