A New Scheme for Fault Tolerant Traffic Grooming in WDM Optical Networks

By

Quazi R. Rahman

A Thesis
Submitted to the Faculty of Graduate Studies
Through the School of Computer Science
In Partial Fulfillment of the Requirements for
The Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2008

Your file Votre référence
ISBN: 978-0-494-42268-7
Our file Notre référence
ISBN: 978-0-494-42268-7

AUTHOR'S DECLARATION OF ORIGINALITY

I, Quazi R. Rahman, hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Traffic grooming techniques in optical networks are attracting increasing research attention in order to handle the huge bandwidth mismatch between high capacity lightpaths and low-rate individual traffic requests. It is important to have guaranteed survivability of all user connections in such networks. There are two popular schemes in use for the protection of WDM networks against any optical link failure – Dedicated Path Protection and Shared Path Protection. However, these schemes require pre-allocating resources for backup lightpaths at design time. Recently, a new scheme has been proposed where every link failure scenario is taken care of when designing a robust logical topology. In our research we have used heuristics to implement this new scheme considering non-bifurcated traffic grooming, and have compared our approach with path protection schemes. Experimental results show that, in respect of network resources utilization, our scheme clearly outperforms both Dedicated Path Protection and Shared Path Protection schemes.

DEDICATION

To my wife
Afshana T. Reshad

ACKNOWLEDGEMENTS

It gives me immense pleasure to thank a few very important people who have helped me a lot during the whole process of my graduate study.

At first, I would like to take this opportunity to express my sincere gratitude to my supervisor, Dr. Subir Bandyopadhyay for his constant guidance and extensive support throughout my graduate studies. This work could not have been achieved without his continuous encouragement, astute advices, suggestions and cooperation.

I would also like to thank Dr. Arunita Jaekel for her valuable advice and suggestions.

I would like to thank all the committee members, Dr. Sazzadur Chowdhury, Dr. Jessica Chen and Dr. Dan Wu for their valuable time, instructive suggestions and comments.

Finally I would like to thank my wife Mrs. Afshana T. Reshad for her endless love and care and for always being there for me.

Last but not the least, I would like to thank everybody who has offered any help during my graduate study at the University of Windsor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

1.0 INTRODUCTION

1.1 Overview

In the past decade we have observed a phenomenal growth of telecommunication networks, which was mostly driven by ever-increasing user demands for new applications as well as continuous advancements in the technologies involved. With the introduction of optical fibers as the data communication medium, which can provide a huge bandwidth capacity, today's optical networks can easily handle the unprecedented bandwidth demand of the modern day communications [3].

However, because of the limited speed of the electronic processing that must be done at the transmitting as well as the receiving ends of any data communication, it is unlikely that the entire bandwidth of an optical fiber will be exploited by using a single high-capacity optical channel or wavelength. For this reason, it is desirable to find an effective technology that can efficiently exploit the huge potential bandwidth capacity of optical fibers.

The emergence of wavelength division multiplexing (WDM) technology has provided a practical solution to meet this challenge. With WDM technology, multiple optical signals can be transmitted simultaneously and independently using non-overlapping carrier wavelengths over a single fiber, each at a rate of a few gigabits per second, which significantly increases the usable bandwidth of an optical fiber [70]. Currently, Dense-

WDM (DWDM) technology can already achieve up to 320 wavelengths per fiber, with each wavelength carrying 10Gb/s, for a total transmission capacity of up to 3.2 Terabits/sec [61].

A WDM optical network consists of a set of end-nodes (any device that produces or consumes data traffic is an end-node) each equipped with optical devices such as optical signal transmitters and receivers, wavelength routers, add/drop multiplexers, optical cross-connects etc[1], and which are interconnected by a set of optical fibers. This configuration defines the *physical topology* [48] of an optical network.

A *logical topology* [48], on the other hand, may be defined over a physical topology by establishing *lightpath*s between the end-nodes [40]. A lightpath is a point-to-point all-optical wavelength channel that connects a transmitter at a source node to a receiver at a destination node to carry encoded optical data from the source to the destination [52]. A lightpath is allowed to pass through any set of intermediate nodes, as necessary, using optical cross connects (OXCs). If a WDM network has no *wavelength converter* at any end-node, a lightpath must be assigned the same channel on all links that it traverses [31]. This is known as the *wavelength continuity constraint* [6]. Due to the limitations of the related technologies and cost involved, most networks today enforce the wavelength continuity constraint and we also follow this restriction in our thesis.

A logical topology consists of the same set of end-nodes as the physical topology and a set of directed edges, called the *logical edges*, connecting those end-nodes. If a lightpath

[1] We have reviewed some of these devices in Chapter 2.

has been established from a source end-node i to a destination end-node j, we say that there is a logical edge from node i to node j. Once the logical topology has been achieved by establishing necessary lightpaths, the physical topology is irrelevant for determining a traffic routing strategy to handle the traffic demand between end-nodes.

The design of a logical topology involves determining the set of lightpaths needed to meet the traffic demands between pairs of end nodes, appropriate routing of the lightpaths over the physical topology (known as the Routing and Wavelength Assignment (RWA) problem) and the proper routing of traffic demands over the logical topology. For a given physical network and the set of lightpaths to be established, the RWA problem is to select a suitable path and a wavelength among the many possible choices for each connection so that no two lightpaths sharing a link are assigned the same wavelength [42].

The main objective of RWA is to design a logical topology such that:

- it can accommodate all the lightpaths using the underlying physical topology,
- it minimizes the use of the network resources and reduce the overall cost, and
- it allows optimal routings of the traffic demands between all node pairs.

To provide a complete solution of the above design problems, many authors have proposed mixed integer linear program (MILP) formulations, [33], [38], [48]. The problem with such formulations is that the formulations are very complex and become computationally intractable even for moderate sized networks [24]. For better handling of

the problem, most existing approaches separate the problem into three independent problems:

1) the logical topology design
2) the RWA for each lightptah in the logical topology and
3) the optimal routing of the traffic over the logical topology.

Typically, heuristics are used to design the logical topology and either an LP formulation or a heuristic is used to determine the routing of the traffic over the logical topology.

1.2 Motivation

Since a single fiber in an optical network can carry tens to hundreds of gigabits of data per second, a huge amount of data is affected in the event of a network failure. Even a momentary disruption of traffic flow may lead to a significant amount of data (and hence revenue) loss, making it a very serious matter [69]. The survivability of WDM networks is a very important issue and, in recent years, there has been intensive research interest in the area of survivable WDM network design [1], [2], [7], [21], [23], [37], [43], [59], [60], and [71].

There may be different types of network failure, but only two types among them are most common and most serious: link failure and node failure. Link failure means the fibers between the nodes in the physical layer are damaged, causing all the lightpaths between the nodes totally disconnected or unusable. Node failure means a workstation or a concentrator in the physical layer is damaged or unable to work due to power failure, fire

or for some other reason. The most common type of failure in WDM networks is the link failure [71].

In this thesis, we propose a new methodology for the design of fault tolerant logical topologies in WDM optical networks, based on the concept of survivable routing [52]. In our approach, we route the lightpaths over the physical topology in such a way that the logical topology remains connected even in the event of a link failure. We also guarantee that the surviving logical topology, after any single link failure, will have sufficient reserve capacity to accommodate the entire traffic demand.

1.3 Problem Statement

The logical topology design problem assumes that the network traffic demands and an underlying physical network are given. To accommodate the traffic demands efficiently, an optimal logical topology and RWA are needed to be determined. The information needed for the design includes the topology of the physical network, the characteristic of the fiber (i.e. number of channels that are available on each fiber) and the number of transmitters and receivers available in the network.

The logical topology design problem for WDM networks consists of two components:

(i) determining a set of logical edges that can accommodate the given traffic demand, and

(ii) determining a physical route and a channel for the lightpath(s) associated with each logical edge.[2]

These two components are inter-related. MILP formulations [33] and heuristics [48] have been proposed for solving the above two components together. In order to reduce the complexity of the problem, the two components are often treated separately. If the set of lightpaths is given, the topology design problem is reduced to the pure RWA problem, which in turn can be broken into the routing sub-problem and wavelength assignment sub-problem.

In our study we assume that the traffic demand is relatively stable and does not change quickly with time. Since the occurrence of simultaneous, or even near simultaneous, multiple physical link failure is relatively rare in optical networks [52], most work in WDM fault tolerance assume single link failure scenarios. Our experimental study assumes single link failure even though we have discussed how the work may be easily extended to handle multiple failures.

With the growth of WDM technology, potentially hundreds of wavelength channels are available in a single fiber. Therefore, wavelengths are no longer scarce resources to be minimized, although a minimal use of it contributes to the minimal use of network resources. The cost of transmitters and receivers at each node is becoming the most important factor determining the cost of a network. Consequently, an efficient logical topology design strategy should try to minimize the number of transmitters and receivers

[2] This is known as the *Routing and Wavelength Assignment* (RWA) problem.

required. This can be achieved by minimizing the number of lightpaths used in a network, since each lightpath requires one transmitter and one receiver.

In our thesis, we have considered the problem of fault tolerant logical topology design as follows:

Given a WDM network and a traffic demand matrix, the problem is to determine a logical topology (i.e. set of lightpaths) such that:

- the cost of the topology is minimized
- the topology remains connected and is capable of handling the entire traffic demand, under all single-link failure scenarios
- a feasible RWA can be determined for the set of lightpaths, over the underlying physical network

In our scheme, there is no need to allocate resources for backup paths, as needed in protection schemes, and there is no need to establish new lightpaths after a fault is detected, as needed in the restoration schemes. Using our approach, the recovery from a failure can be achieved simply through traffic rerouting. The logical topology derived using our heuristic may not always be the best one, but we guarantee that it is feasible with respect to the given physical network. For every link failure scenario, our heuristic also determines a strategy to route the traffic over the remaining logical edges in such a way that the capacity of a lightpath is never exceeded.

One of the main objectives in our heuristic is to reduce the number of lightpaths in the logical topology as much as possible. The robust logical topology, generated using our heuristic, of course, requires some additional resources (in terms of the number of lightpaths and the sum of the number of carrier wavelengths on the fibers in the network) compared to a network with no provision to handle faults. We have tested our approach on a number of randomly generated physical networks and have compared our results with survivable topologies based on existing path protection schemes. The results demonstrate that our approach is much more efficient in terms of resource utilization.

1.4 Organization Of Thesis

We have organized the remainder of the thesis as follows. In Chapter 2 we have reviewed some relevant topics in WDM networks and logical topology design. In Chapter 3 we have presented our heuristic algorithm for fault tolerant logical topology design and also have discussed the implementation details of the heuristic. In Chapter 4 we have presented the experimental results on some randomly generated networks. Finally, in Chapter 5 we have concluded our thesis with some remarks and directions for future works.

CHAPTER TWO

2.0 REVIEW OF RELATED TECHNOLOGY

2.1 Optical Technology

In this section we discuss some of the hardware devices used in optical networks, including Optical Fibers, Optical Signal Amplifiers, Optical Add-Drop Multiplexers, Wavelength Routers, Optical Transceivers.

2.1.1 Optical Fibers

Optical fibers are long, thin strands of very pure glass about the diameter of a human hair. They are arranged in bundles called optical cables and are used to transmit optical signals over long distances. An optical fiber consists of a cylindrical core of silica (Figure 2.1), with a higher refractive index, surrounded by cylindrical cladding, also of silica, with a lower refractive index [3]. The idea of optical communication using a fiber is that, if an optical signal passing through an optical medium with a higher refractive index, say μ_1, meets another optical medium with a lower refractive index, say μ_2, at an angle greater than the critical angle $sin^{-1}\mu_2/\mu_1$, *total internal reflection* takes place where the signal is entirely reflected back into the denser medium [3]. Optical signal propagates through the core of the fiber using a series of such total internal reflections (Figure 2.2).

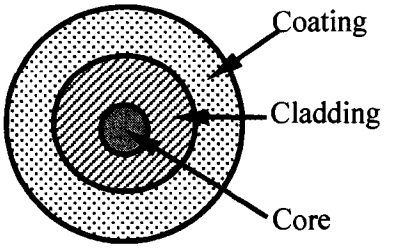


Figure 2.1 Cross-section of an optical fiber

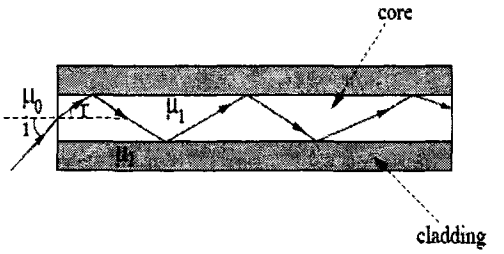


Figure 2.2 Optical signal propagation using total internal reflection

The potential transmission capacity of optical fiber is enormous. Especially, within the last 30 years, the transmission capacity has been increased dramatically. Theoretically, using advanced techniques such as DWDM (Dense WDM), the modest number of fibres as seen in Figure 2.3 (Figure taken from http://en.wikipedia.org/wiki/Optical_fiber) could have sufficient bandwidth to easily carry the sum of all types of current data transmission needs for the entire planet. (~100 terabits per second per fibre). However, the transmission capacity of a fiber is strongly dependent on the length of the fiber. The longer a fiber, the lower is its achievable transmission rate [5].

Currently two types of optical fiber are in use – multimode fibers and single-mode fibers. Multimode fibers are often more convenient to utilize for short distances of a few hundred meters or less, as they are cheaper to install and the necessary equipments are

less expensive. Multimode fibers can achieve data rates between a few hundred Mbps to around 10 Gbps, depending on the transmitter technology.

Figure 2.3 A bundle of optical fibres
(Taken from http://en.wikipedia.org/wiki/Optical_fiber)

Single-mode fibers are typically used for longer distances of a few kilometers or more. Currently a single encoded optical signal can be used to transmit at the rate of 2.5 Gbps, 10 Gbps or even 40 Gbps, over a distance of ten kilometers or more. Future systems may use higher data rates of 160 Gbps. As we explain in Section 2.2.1, the actual data transmission capacity of an optical fiber is much higher.

2.1.2 Optical Signal Amplifiers

While traveling through the optical fiber, a part of any optical signal is always lost due to a variety of factors, including scattering, absorption, and bending, even though the modern day optical fibers are made from extremely pure silica. Signal amplifiers are therefore required, especially for long-haul transmission cables, to boost up the weak signals. There are several different types of optical amplifiers that are being used in

today's optical communication systems. Among them three most widely used are Semiconductor optical Amplifiers (SOA), Erbium-Doped Fiber Amplifiers (EDFA), and Raman Amplifiers [29]. A typical semiconductor optical amplifier (SOA) is a waveguide structure with a semiconductor gain medium [10], similar to a semiconductor laser. Due to their compact size, reduced power consumption and reduced cost of fabrication, semiconductor optical amplifiers are popular for short to intermediate reach, narrow band gain applications [11]. The disadvantages of SOAs include much narrower wavelength bands, reduced amplification, and higher noise figure than erbium-doped fiber amplifiers (EDFA) [58]. EDFAs are widely used in line amplifiers for long distance links and other applications requiring high output power, high data rates, and low noise. An EDFA can simultaneously amplify signal light of multi-wavelengths within an amplification spectrum band. Therefore, it is widely used as an optical amplifier applied to a wavelength division multiplexing (WDM) transmission system. A recent technology uses a circuit of EDFA to fully exploit the spectrum of all-wave fiber and it is called ultra wide-band EDFA [39]. Raman amplification is also becoming increasingly important in optical communication systems, in particular in high-bit rate WDM and DWDM systems. An important advantage of Raman amplification is that the effective optical signal-to-noise ratio is significantly lower than that of an erbium-doped fiber amplifier for the same gain. However, the Raman amplifier not only has very low optical amplification efficiency, but also needs a high-priced pumping light source, thereby increasing the size and the price of the optical amplifier module [58].

2.1.3 Optical Add-Drop Multiplexers (OADM)

An optical add-drop multiplexer (OADM) is a device used for multiplexing and routing different channels of optical signals into or out of an optical fiber in a WDM network system (Figure 2.4). "Add" and "drop" here refer to the capability of the device to add one or more new optical signals using an unused channel to an existing set of WDM signals, each using a different channel, and/or to drop (i.e., remove) one or more optical signals received as the input the device.

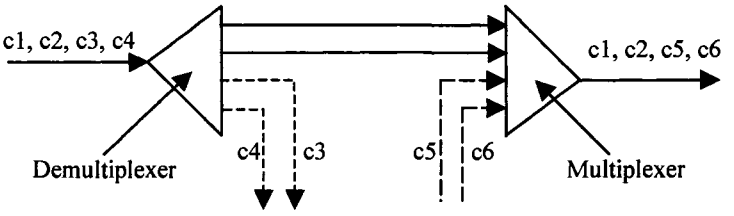


Figure 2.4 Schematic diagram of an OADM

A traditional OADM consists of three stages: an optical demultiplexer, an optical multiplexer, a method of reconfiguring the paths between the optical demultiplexer, the optical multiplexer and a set of ports for adding and dropping signals. The optical demultiplexer separates the signals on the input fiber, using different wavelengths and directs them to the optical multiplexer or to the drop ports as it has been configured. The optical multiplexer combines the incoming optical signals, which are not routed to the drop ports, with the signals received at the add ports, onto a single output fiber.

2.1.4 Wavelength (Lambda) Routers

Wavelength routers (Figure 2.5) - which are also called *Lambda Routers*, or *Optical Cross-Connects* (OXC) - are normally positioned at network junction points or router nodes. The lambda router takes in a single wavelength of light from a specific optical fiber and recombines it into another fiber that is set on a different path, without going through any opto-electronic conversion.

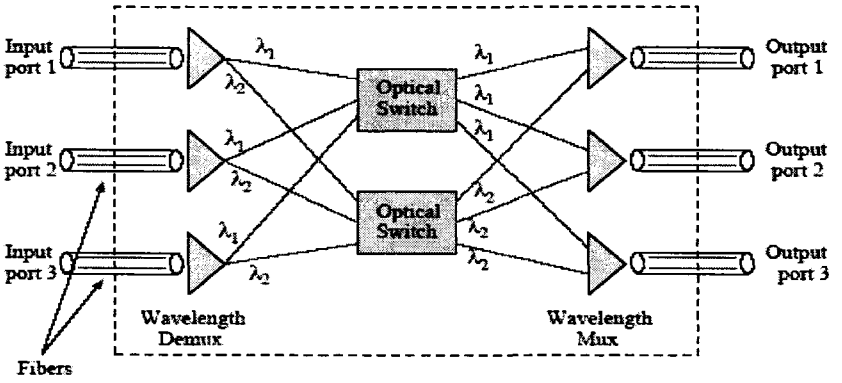


Figure 2.5 Wavelength Router

2.1.5 Optical Transceivers

An optical transmitter is a device that accepts an electronic signal as its input, processes this signal, and uses it to modulate an optoelectronic device, such as an LED or an injection laser diode, to produce an optical signal capable of being transmitted via an optical transmission medium [22]. An optical receiver is a device that accepts an optical signal as its input, processes this signal through an electro-optical device to convert it into an electronic signal to be further processed by electronic devices.

Fiber optic transceivers include both a transmitter and a receiver in the same component. These are arranged in parallel so that they can operate independently of each other. Both the receiver and the transmitter have their own circuitry so that they can handle transmissions in both directions.

2.1.6 Wavelength Converters

Wavelength converters are devices used at the router nodes of WDM or DWDM networks such that a lightpath traveling through multiple fiber links can be assigned different wavelength channels in different links. Using wavelength converters in a network is expensive, but it can eliminate the wavelength continuity constraint of a lightpath so that, in a network with wavelength converters at each node, a lightpath may be assigned different channels on successive fibers used in its route, which greatly reduces number of required channel in a fiber.

2.2 WDM Optical Network

As we have mentioned before, a single optical fiber has, at least theoretically, a potential bandwidth of nearly 50 terabits per second (Tbps), which is about four orders of magnitude higher than the currently achievable electronic processing speed of a few gigabits per second (Gbps) [3]. However, because of the limits of the electronic processing speed, it is unlikely that all the bandwidth of an optical fiber can be exploited by using a single high capacity optical channel. For this reason, it is desirable to find an effective technology that can efficiently exploit the huge potential bandwidth capacity of optical fibers. The emergence of wavelength division multiplexing (WDM) technology

has provided a practical solution to meeting this challenge. With WDM technology, multiple optical signals can be transmitted simultaneously and independently in different optical channels over a single fiber, each at a rate of a few gigabits per second, which significantly increases the usable bandwidth of an optical fiber [70].

Besides the increased usable bandwidth of an optical fiber, WDM also has other advantages, such as, efficient failure handling, which means we can overcome more efficiently the data communication interruption due to any failure of communication media or the related software, data transparency, means data are more reliable and fault-free, and also reduced electronic processing cost [3]. As a result, WDM has become the technology of choice to meet the tremendous bandwidth demand in current and future networks. Optical networks using WDM technology are being considered as the potential main network infrastructure for the next-generation of telecommunications networks and the Internet [70]. In the following sections we have discussed some of the important aspects of WDM technology.

2.2.1 WDM Technology

Wavelength division multiplexing (WDM) is an optical multiplexing technology to use the huge bandwidth capacity of the optical fibers. It is conceptually similar to frequency modulation (FM) that is being used in radio communication systems for over a century. The basic principle is to divide the huge bandwidth of an optical fiber into a number of non-overlapping sub-bands or optical channels and transmit multiple optical signals simultaneously and independently in different optical channels over a single fiber [3].

The attenuation of an optical signal propagating through a fiber is acceptably low (around 0.2 dB/km) in the wavelength band of 1450 to 1650 nanometers (nm). One is centered at 1300 nanometers (nm) and the other at 1500 nm. Within this interval, the band from wavelengths 1530 to 1565 nm is called the *C-band* (conventional band) and is widely being used for optical communication in WDM networks. Most of the optical devices for this band are currently available in the market. Other bands such as the *L-band* (long band) from 1565 to 1625 nm are expected be available in the near future [70].

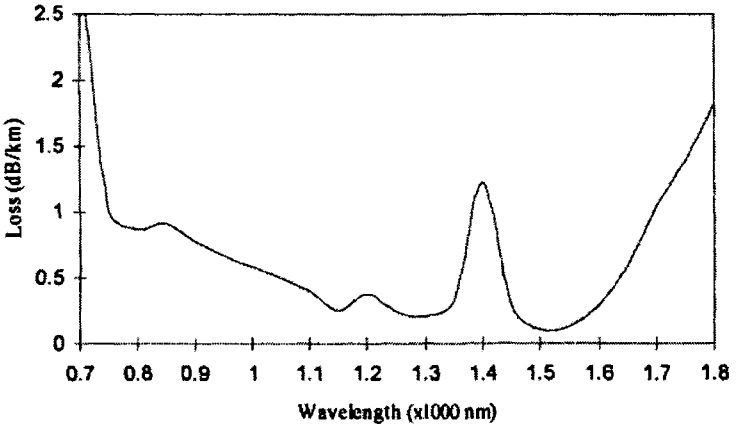


Figure 2.6 Transmission spectrums of optical fibers.

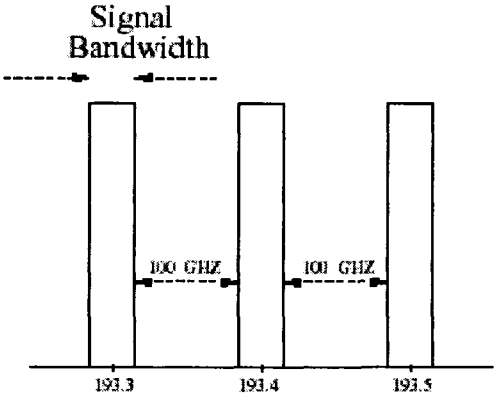


Figure 2.7 Signal bandwidth and channel spacing.

Generally, an optical fiber carries a number of optical signals in the band being used. These optical signals must obviously be at different carrier wavelengths. It is convenient to visualize the available bandwidth (which is currently the C-band) as a set of ranges of wavelengths, or *channels*, as they are usually called. Each signal is allotted a distinct channel such that each channel has a bandwidth to accommodate the modulated signal. In order to avoid any interference between different signals, each channel is separated from every other channel by a certain minimum bandwidth called *channel spacing* (Figure 2.7). Typically, a channel bandwidth of 10 GHz and a channel spacing of 100 GHz are currently being used. This means that the C-band can accommodate up to 80 channels, each having a bandwidth of 10 GHz. Shorter channel spacing (25 GHz) will lead to as many as 200 channels in the C-band alone.

The potential data carrying capacity of a single modern day optical fiber is around 50 Tbps (Tera bits per second) or beyond. Recently, Nippon Telegraph and Telephone Corporation of Japan have announced that they have successfully demonstrated the ultra-large capacity optical transmission of 14 Tbps (using 140 channels, each at a speed of 111 Gbps) [74]. Figure 2.8 below shows a graph depicting the future trend of data carrying speed of optical fibers (Figure taken from http://www.fiber-optics.info/fiber-history.htm)

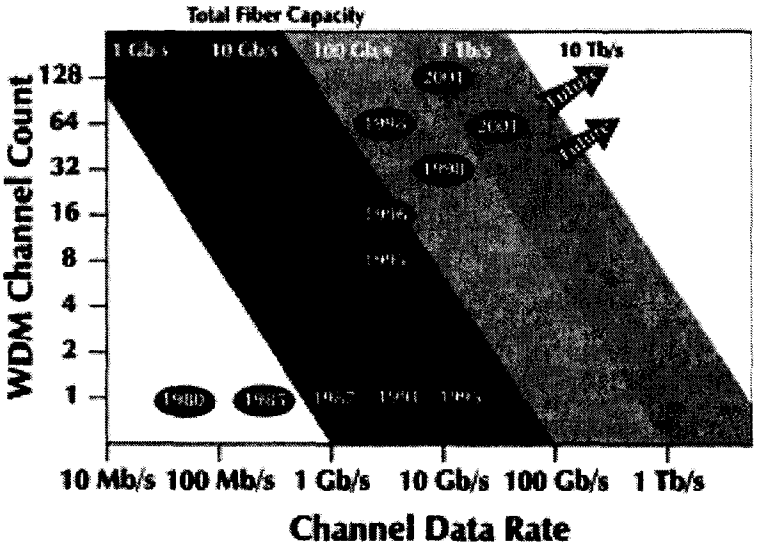


Figure 2.8 Future Trend of Data Rate in Optical Fibers
(Taken from http://www.fiber-optics.info/fiber-history.htm)

2.2.2 Faults in WDM Networks

As WDM optical networks are becoming more and more popular for today's fast telecommunication networks and the Internet, the demand for a fault free or fully fault tolerant network system is also increasing. Since a huge amount of data can travel at a tremendous speed through the fibers of the optical networks, even a momentary interruption of any component of the network system can cause the loss of a large amount of data.

As optical networks are being rapidly deployed on a global scale, which involves millions of kilometers of optical fibers and thousands of other network components, protecting a network from different types of faults and failures have become particularly important.

One of the major challenges in design and maintenance of the today's large-scale optical networks is the survivability of the network.

In a WDM network, failure may occur in any component of the network. This includes link failures, node failures, channel failures and/or software failures. Link failure is the most common type of fault where the fiber constituting a link between two nodes in the network does not permit data transmission. Since a single fiber can carry 100 or more lightpaths, and each lightpath can carry data at the rate of 2.5 Gbps to 10 Gbps, even a brief disruption of this traffic is a serious matter [3].

2.2.3 Survivability of WDM Networks

There are two basic ways of fault recovery to ensure optical network survivability [68]. One is known as the *protection* scheme based on dedicated resources and the other is known as the dynamic *restoration* scheme. In the dedicated resource-based protection scheme, the network resources may be either dedicated for each failure scenario, or may be shared among different failure scenarios. In the dynamic restoration scheme, the spare capacity available within the network is utilized for restoring services affected by a failure. Generally, dynamic restoration schemes are more efficient in utilizing the capacity of the network due to the multiplexing of the spare-capacity requirements and provide resilience against different kinds of failures, while dedicated-resource protection schemes have a faster restoration time and provide guarantees on the restoration ability [68]. A categorization [44] of fault management scheme for WDM optical network is depicted in the figure 2.9 below.

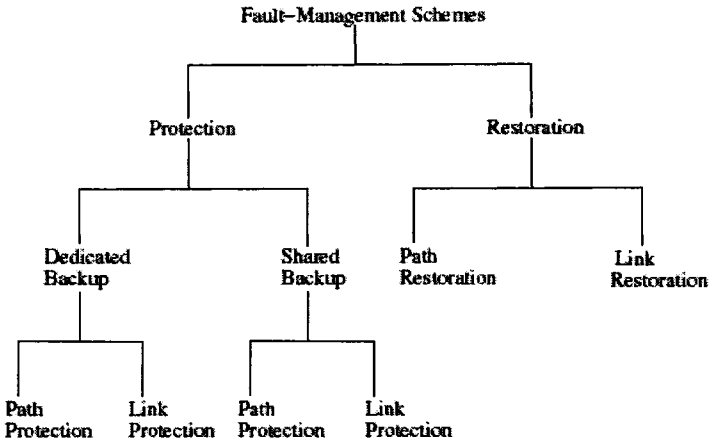


Figure 2.9 A categorization of fault management schemes

There are two major techniques that are in use to handle link failures in optical networks:

(i) protection based techniques [12], [18], [54], and

(ii) restoration based techniques [9], [13], [14], [35], [47].

Protection-based techniques are based on the provisioning of backup paths to recover from a failure. During the period of establishing lightpaths, network resources are kept reserved, such that, when a failure occurs, data can be rerouted around the affected links/lightpaths. In a traditional path protection scheme, if a logical edge is established from node *i* to node *j*, then two lightpaths are actually set up. The first one, called the *primary* lightpath, carries the data under normal fault-free conditions and the second one, called the *backup* lightpath, which is link-disjoint with respect to the primary lightpath, carries data only when the primary lightpath fails. In case of a network failure, such as a broken link on the primary path from node *i* to node *j*, the primary lightpath from node *i* to node *j* is disrupted. In this situation the data from node *i* to node *j* are sent through the corresponding backup lightpath. Since a primary lightpath and the corresponding backup

lightpath are link-disjoint, there will always be a valid lightpath from node *i* to node *j*, for any single link failure scenario. This approach is more efficient in respect of response time, but the drawback of this approach is that the resources allocated to the backup paths remain idle, and are wasted under normal conditions.

Restoration-based techniques, on the other hand, dynamically search for the spare capacity in the network to establish new lightpaths in order to restore the affected services *after* a network failure is detected [47]. There is no allocation of resources for backup paths at design time. Such techniques are more efficient in terms of resource utilization. However, restoration takes longer time than protection to restore services (since backup paths are not known in advance) and there is no guarantee that all the affected lightpaths can be restored. In summary, both protection and restoration schemes require the setting up or the creation of new lightpaths, when a fault is detected.

2.2.3.1 Path Protection

In path protection, backup resources are reserved during connection setup. When a link fails (Figure 2.10(a)), the source node and the destination node of each lightpath that traverses the failed link are informed about the failure via messages from the nodes adjacent to the failed link, as illustrated in Figure 2.11.

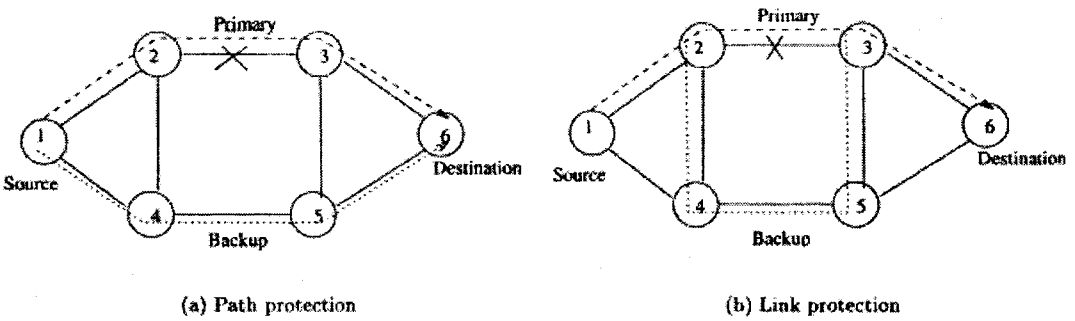


Figure 2.10 Protection Schemes

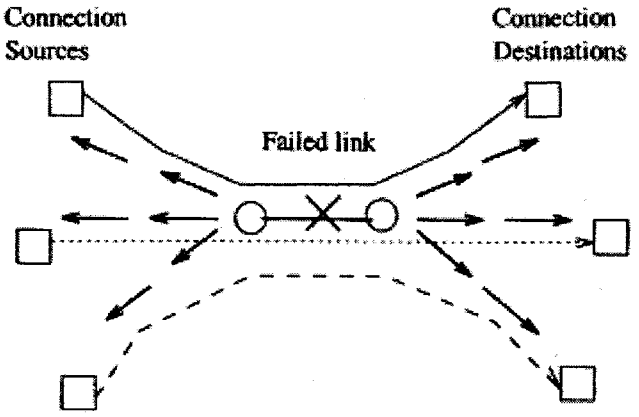


Figure 2.11 Link-fail messages.

Path protection scheme is sub-divided into two categories [52] – *dedicated* path protection and *shared* path protection. In dedicated path protection, the resources along a backup path are dedicated for only one lightpath and are not shared with the backup paths for other lightpath. Dedicated protection scheme may be further classified as 1+1 or 1:1. In 1+1 protection scheme, the data is sent *simultaneously* using both the primary and the backup lightpaths. If there is a fault in the primary path, the destination simply continues to get the data from the backup path. The recovery of the network from faults in such a scheme is very fast since the destinations affected by a fault do not need to communicate

with the corresponding sources or have to wait for any corrective action. In 1:1 protection scheme, when the network is fault-free, the data is sent using only the primary lightpath. If there is a fault in the primary path, the destination node informs the corresponding source node, which then stops communicating using the primary path, sets up the backup lightpath using the backup path, and then continues data transmission using the lightpath. The recovery of the network from faults in 1:1 scheme is slower than the 1+1 scheme since the destinations affected by a fault need to communicate with the corresponding sources, wait for the source to set up the backup lightpath, and then to communicate over to the backup lightpath. There is one advantage of the 1:1 scheme over the 1+1 scheme. Since the backup lightpath is not set up until there is a fault and the probability of a fault is low, the resources for the backup lightpaths are not needed most of the time. These resources may be used for low priority traffic until there is a fault and there is a need to set up a backup lightpath.

In dedicated protection approximately 50% of network resources are allotted to backup lightpaths. Since faults occur rarely, these resources are wasted, in the sense that they are not used most of the time. Shared path protection (also called *backup multiplexing* [20] or *1:N protection*) reduces this wastage to some extent. The idea is that, if two primary paths use edge-disjoint lightpaths, then, under the single fault assumption, both the primary paths can never contain a faulty edge simultaneously. Therefore it is never necessary to use the respective backup lightpaths at the same time. In shared-path protection, the resources along a backup path may be shared with other backup paths. As a result, backup channels are multiplexed among different failure scenarios (which are

not expected to occur simultaneously), and therefore, shared-path protection is more capacity efficient when compared with dedicated-path protection. However some time is needed to configure the routers on the backup path before the backup lightpath may be set up.

2.2.3.2 Path Restoration

In path restoration [68], the source and destination nodes of each connection traversing the failed link participate in a distributed algorithm to dynamically discover an end-to-end backup route. If no routes are available for a broken connection, then the connection is dropped. This scheme may be resource efficient, but at the same time it is time consuming. After a fault occurs, the system has to search the entire network to find a suitable path (which, sometimes, may not be possible) to restart sending data. This scheme is not desirable for high priority data.

2.2.3.3 Link Protection

In link protection, backup resources are reserved around each link during connection setup. In link protection/restoration [Figure 2.10(b)], all the connections that traverse the failed link are rerouted around that link, and the source and destination nodes of the connections are oblivious to the link failure.

Link protection scheme can also be sub-divided into two categories – *dedicated* link protection and *shared* link protection [44]. In dedicated link protection, at the time of connection setup, for each link of the primary path, a backup path and a wavelength are

reserved avoiding that link and are dedicated to that connection. In practice, it may not be possible to allocate a dedicated backup path on each link of the primary connection and on the same wavelength as the primary path. It has been found that dedicated-link protection utilizes wavelengths very inefficiently, and therefore, is not popular.

In shared link protection, the backup resources reserved along the backup path may be shared with other backup paths. As a result, backup channels are multiplexed among different failure scenarios (which, as in the earlier case, are not expected to occur simultaneously), and therefore shared-link protection is more capacity-efficient when compared with dedicated-link protection.

2.2.3.4 Link Restoration

In link restoration, the end nodes of the failed link participate in a distributed algorithm to dynamically discover a route around the link. If no routes are available for a broken connection, then the connection is dropped.

As discussed above, though restoration schemes are more network capacity efficient, the uncertainty of recovering from a fault and slow recovery has made this scheme unpopular. Similarly link protection/restoration schemes have also some drawbacks as mentioned above. All these facts have made path protection schemes more popular for making a network survivable, which give a fast and guaranteed recovery from a fault, may be with some price tag. Most of the research works on survivable WDM network has been carried out considering path protection scheme.

2.3 Traffic Grooming in Optical Networks

As mentioned before, the transmission capacity of a fiber in today's optical networks has increased significantly due to wavelength-division multiplexing (WDM) technology. Each lightpath in a WDM network carries 10 Gbps or 2.5 Gbps depending on the technology used. The network performance is now mainly limited by the processing capability of the network elements, which are mainly electronic. Moreover, individual requests for connection are typically for much lower data communication rates, of the order of Mbps. By efficiently grooming low-speed traffic streams onto high-capacity optical channels, it is possible to minimize this electronic processing and eventually increase the network performance. Traffic Grooming in WDM can be defined as a family of techniques for combining a number of low-speed traffic streams from users so that the high capacity of each lightpath may be used as efficiently as possible. Traffic grooming minimizes the network cost in terms of transceivers and optical switches [3].

"Traffic grooming is composed of a rich set of problems, including network planner, topology design, and dynamic circuit provisioning" [53]. The traffic grooming problem based on static traffic demands is essentially an optimization problem. It can be seen as a dual problem from different perspectives. One perspective is that, for a given traffic demand, the design has to satisfy all traffic requests as well as to minimize the total network cost. The other problem is that, for given resource limitation and traffic demands, maximize network throughput, i.e., the total amount of traffic that is successfully carried by the network [68]. In recent years, there has been an increasing amount of research activity on the traffic grooming problem, both in academia and in the

industry. Researchers are realizing that traffic grooming is a practical and important problem for WDM network design and implementation.

2.4 Terminologies Used in WDM Networks

In this section we discussed some terminology and concepts that are well known to optical networks and network design.

2.4.1 Physical Topology

The physical topology of a network may be simplified for our discussions and may be represented by a graph where each end node (or router node) in the network is represented as a vertex in the graph and each fiber optic link between two nodes is represented as an edge [51]. This edge in the graph is commonly known as the *physical link*, or simply a *link*. Each fiber link is usually bi-directional, therefore the graph is assumed to be undirected.

2.4.2 Lightpath

A lightpath is defined as an all-optical channel between two nodes in which traffic will not be converted into electronic forms at any intermediate node [51]. Traffic signal remains and is routed as optical signal throughout the length of the lightpath. In a network where the wavelength continuity constraint is satisfied, each lightpath may pass through any number of physical links and has to use on the same channel on all fiber links that it traverses. Thus, two lightpaths that share a common fiber link should not be assigned the same wavelength. If each switching/routing node is also equipped with a

wavelength converter, then a lightpath may use different channels on different fiber links on its route from origin to termination.

2.4.3 Logical Topology

A logical topology is also a graphical representation of an optical network where the vertices are the same set of nodes as in the physical topology, but where the edges are the set of lightpaths [51]. Since a lightpath is always one directional, the logical topology of network is represented by directional graph. An example of a physical topology and corresponding logical topology is depicted in figure 2.12 below.

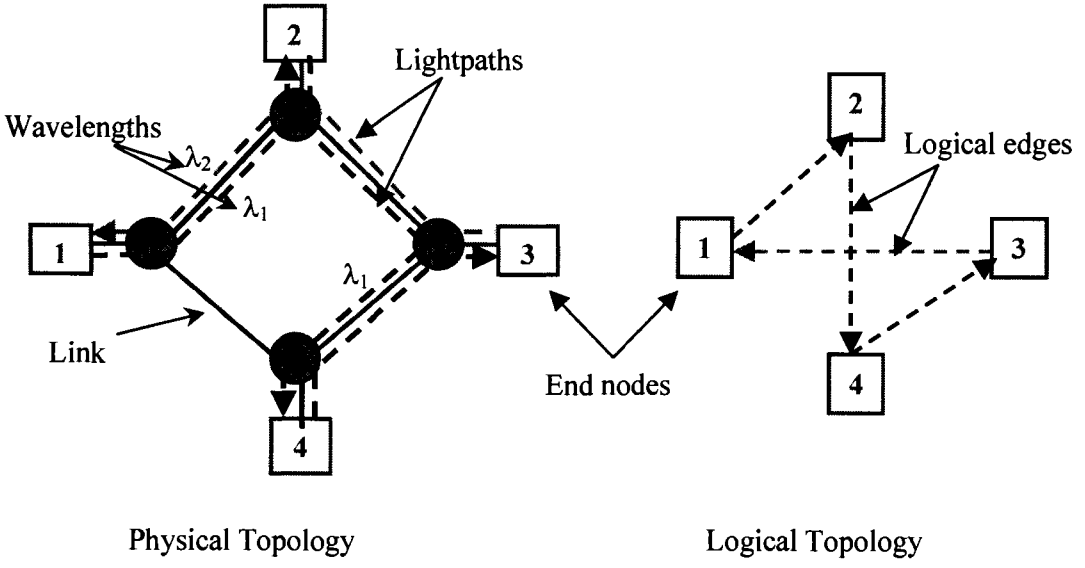


Figure 2.12 Physical and Logical Topology

2.4.4 Wavelength Continuity Constraint

In a WDM network, end nodes communicate with each other using modulated optical signals that are commonly known as lightpaths. Lightpaths are used to support optical

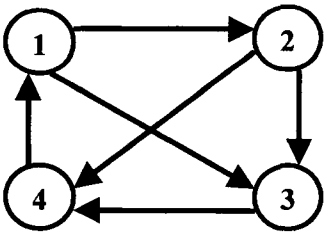
connections and may span multiple fibers in a WDM network. A fiber link normally supports many channels, each corresponding to a unique range of wavelengths, from which every lightpath has to be assigned a range of wavelength. In this regard, normally a lightpath operates on the same channel across all fiber links that it traverses, in which case the lightpath is said to satisfy the wavelength continuity constraint [6]. However, if the switching/routing nodes are equipped with wavelength converters, then a lightpath may switch between different wavelengths on its route from its origin to its termination and the wavelength continuity constraints is not applicable.

2.4.5 Routing and Wavelength Assignment (RWA)

The logical topology in a WDM optical network is defined using a set of logical edges or lightpaths. To establish a lightpath, it is important to find a suitable route for it in the physical topology and assign a channel to it for every fiber in its route. Given a physical topology and a set of connection requests, the problem of setting up of lightpaths and assigning channels to each of these lightpaths is known as Routing and Wavelength Assignment (RWA) problem [66]. In a network where no wavelength converter is available, a lightpath must be assigned the same channel on all the fiber links it traverses, satisfying the wavelength continuity constraint. In networks with full wavelength converters at each node, the channel used by a lightpath may vary from one fiber to another.

2.4.6 Traffic Matrix

Traffic Matrix specifies the amount of traffic, using some convenient unit to represent data transmission rates, to be transmitted between each pair of nodes in the network [17]. If there are N nodes in a network, the corresponding traffic matrix is an $N \times N$ matrix and denoted by $T = \{t_{sd}\}$, where t_{sd} is the traffic request from node s to node d.



Logical topology
(a)

Node	1	2	3	4
1	0	OC-3	OC-48	OC-3
2	0	0	OC-6	OC-12
3	0	OC-3	0	OC-48
4	OC-24	OC-6	OC-6	0

Traffic Matrix
(b)

Figure 2.13 4-Node logical topology and corresponding traffic matrix.

Figure 2.13(a) illustrates an example of a logical topology of WDM network with 4 end nodes and 2.13(b) shows a possible traffic matrix for such a network. All elements lying on the diagonal of the traffic matrix must be zero, since no traffic can flow from a node to itself. The unit for data communication rate is usually expressed using gigabits/second (Gbps), using a percentage of the data transmission rate of a single lightpath (typically either 2.5 Gbps or 10 Gbps) or using the Optical Carrier notation (i.e., OC-n notation) [73]. In this notation the speed of data is given by n x 51.8 Mega bits per second. For example, OC-24 is equal to 24 x 51.8 = 1243.2 Mbps, which is equivalent to 1 Giga bits per second (1 Gbps). In the above example of traffic matrix shown in Figure 2.13(b), we have expressed the traffic using OC-n notation. For example, the traffic from node 1 to

node 4 is OC-3 means that amount traffic request is around 155 mbps (equivalent to 150 mbps).

2.5 Related Works

Traffic grooming techniques in WDM networks combine a number of low-speed traffic streams from different users so that the high data rate of the lightpaths may be used as "efficiently" as possible [16], [25], [26], [30], [56], [65], [67], [72]. Traffic grooming can use either the bifurcated model or the non-bifurcated model [17]. In the non-bifurcated model, each data stream for a user is communicated, using a single logical path from the source of the data stream to its destination. In the bifurcated model, each user data stream is communicated using one or more logical path(s) from the source of the data stream to the destination. In other words, in the non-bifurcated model, whenever there is a user request for communication from a source end-node to a destination end-node, the data stream corresponding to request becomes part of the payload of each lightpath in the selected logical path. This model has been adopted in [19], [26], [65]. In the bifurcated model, the data stream corresponding to any user request for data communication is allowed to split into an arbitrary number of data streams at any intermediate point, where the resulting data streams, each having a lower data communication rates than that of the request, is carried by a logical path from source to destination. This process of splitting may occur multiple times as needed. The bifurcated model allows more efficient use of network resources but the non-bifurcated model has a number of technological advantages [65].

The traffic grooming problem in WDM optical networks may be viewed as depicted in the figure 2.14 below [27]. User's traffic requests are being groomed over a logical (or some time known as virtual) topology, which is basically a set of logical edges. These logical edges are being efficiently routed, and assigned appropriate wavelength over a physical network.

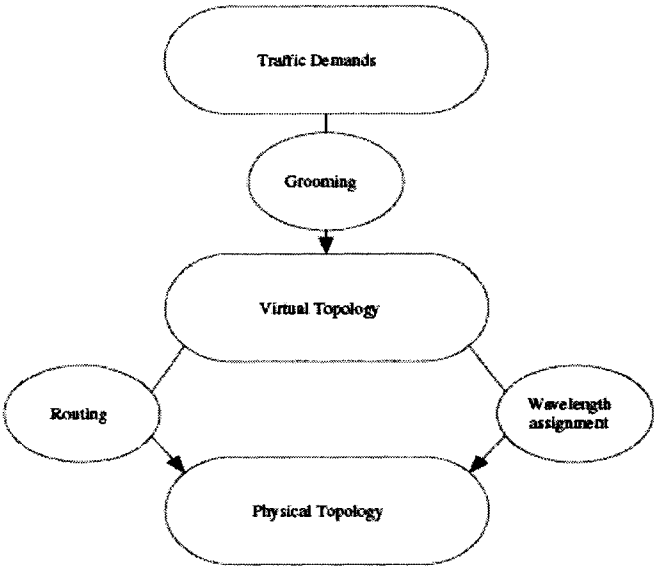


Figure 2.14 A view of the Traffic Grooming Problem

In this section we have discussed some of the significant work done in this field over the past few years.

The authors of the paper [8] address the problem of traffic grooming for unidirectional SONET/WDM ring networks and have developed algorithms for the purpose. In the paper [4], the authors address the problem of minimizing the number of expensive Add-Drop multiplexers (ADMs) in a SONET/SDH optical ring network by proposing the Reactive Local Search (RLS) heuristic. The problem of traffic grooming to reduce the

number of transceivers in optical networks has been studied by the authors of the paper [32]. The authors of the paper [16] address the problem of traffic grooming for wavelength-routed optical networks. The problem addressed by the authors of the paper [36] is the reconfiguration of wavelength routed optical networks in the context of groomed sub-wavelength traffic. According to the authors, it is widely recognized that grooming of sub-wavelength traffic into the full-wavelength channels is an indispensable component of optical network design, but has received comparatively little attention.

In the paper [57] the authors addressed the problem of traffic grooming in WDM mesh network by proposing a new capacity correlation model to compute the blocking performance on a multi-hop single wavelength path. The authors of the paper [63] present a study of the performance analysis of the multi-hop online traffic grooming algorithm in mesh WDM optical networks. The authors of the paper [34] propose a new Genetic Algorithm (GA) – a search tool, to handle the traffic grooming problem in WDM optical networks, extending the classical GAs with heuristic approach to support network cost optimization for combining multiple traffic streams into a single lightpath. In [72], the authors propose a new generic graph model to solve the problem of traffic grooming in heterogeneous WDM mesh networks using various grooming policies and traffic-request-selection schemes.

The authors of the paper [27] identify the various problems that are being faced for grooming dynamic traffics in WDM optical networks to minimize the network cost, and indicate various aspects of problems of this area. In [64], the authors address the problem

of dynamic traffic grooming in WDM mesh networks by first designing a static logical topology a priori based on estimated traffic loads, and then routing each dynamically arriving client call on the established logical topology. The authors of the paper [56] address the problem of dynamically establishing dependable low-rate traffic stream connections in WDM mesh networks with traffic grooming capabilities.

The authors of the paper [19] address the problem of enabling traffic grooming capability in the design of survivable WDM mesh networks. According to the authors, this paper deals with lightpath protection schemes for sub-wavelength level traffic grooming networks, which are defined as shared-wavelength grooming networks with wavelength continuity constrained grooming nodes. The authors of the paper [28] propose protection for multi-granular optical networks against near-simultaneous dual-failures using capacity re-provisioning. They also present a study on the performance of re-provisioning under two different protection frameworks – lightpath level protection and connection level protection.

In the paper [41], the authors investigate the problem of survivable traffic grooming for optical mesh networks that employs wavelength division multiplexing. The authors of the paper [65] present a study of survivable traffic grooming problem in WDM mesh optical networks employing path protection at the connection level. In the paper [61], the authors address the problem of dependable traffic grooming of low-rate connections in WDM mesh networks. They have presented a Shared Protection Traffic Grooming algorithm based on wavelength layered-graph (SPTG-LG). The authors of the paper [62] address

the problem of survivable traffic grooming in WDM mesh networks, by proposing a differentiated shared protection algorithm called, Partial Shared-path Protection algorithm supporting Traffic Grooming (PSPTG). The authors state that not much research has been done on survivable low rate connection traffic grooming in WDM mesh networks.

In the case of non-bifurcated traffic grooming using path protection, each request is allotted to a single logical path. Since each primary lightpath has a corresponding backup lightpath and the route of the backup lightpath is edge-disjoint with respect to the route used by the primary lightpath, each logical edge is robust against single link failure. In other words, the logical topology itself is robust with respect to single link failures. This approach is used in [20], [53]; where the concept of backup multiplexing is extended to include cases where primary paths may not be edge-disjoint. If two primary paths are not edge-disjoint, the corresponding backup paths are still allowed to share all or some links and channels provided sufficient bandwidth exists in the backup paths.

Our approach does not use any "protection backup path", but ensures that there are sufficient spare capacities in the logical topology to re-route all the affected traffic for any single link failure scenario. This is somewhat similar to the concept of "L+1 fault tolerance" discussed [20], [46]. However, these papers deal with dynamic lightpath allocation, where requests are processed one at a time and do not consider integrated optimization of over all traffic requests. Furthermore, the connection requests are at the lightpath level, so sub-wavelength traffic grooming is not an issue in the above papers.

CHAPTER THREE

3.0 HEURISTIC FOR SURVIVABLE TRAFFIC GROOMING

3.1 Overview

In this chapter, we describe our approach for designing a survivable logical topology. The logical topology, generated using our approach, always has a survivable routing. In other words, the logical topology remains connected for all possible single link failures. Furthermore, we guarantee that the surviving topology is not only connected, but is capable of handling the entire traffic demand for any single link failure scenario. In this chapter, we first define the WDM network model used in our research, which includes the physical topology, the network characteristics, and the traffic matrix. Then, we discuss the objectives of our algorithm and finally we give a detailed description of our strategy for designing a survivable logical topology.

3.2 WDM Network Model Definition

The WDM network model used in our design is a mesh topology consisting of a number of end nodes (each end node is associated with a router node), and a number of physical links connecting two end nodes. Each link between end node i and end node j is bi-directional and consists of two unidirectional fibers. We assume that the capacity of each fiber (i.e. the number of WDM channels available on the fiber) and the capacity of a single WDM channel of each fiber (i.e. the maximum amount of traffic it can carry) is the same.

In this thesis we have expressed the data communication rate or data carrying capacity using the OC-n notation [3].

We are given the following information about the current state of network:

- A physical optical fiber network represented by a directed graph $G_P = (V_P, E_P)$ with $|V_P| = N$ (the number of end nodes), and $|E_P| = m$ (the number of physical links).
- A traffic demand matrix $T = \{t_{sd}\}$, where t_{sd} is the total amount of traffic requests between a source node s and a destination node d. We assume that t_{sd} is expressed in OC-n notation with values varying from OC-3 to OC-24.

Before running the simulation test using our algorithms, we have assumed that:

1. Each fiber in the network can accommodate K channels. We have selected $K = 64$.

2. We pre-calculate R edge disjoint physical paths between each source-destination pair. Our algorithms search for the best usable path among these R paths to establish a new lightpath. We have selected $R = 3$. This is in line with the values reported in [45].

3.3 Objective

Given the physical topology of an optical network and a traffic demand matrix, the goal of our study is to determine a logical topology (a set of logical edges) to handle the entire traffic communication requirements specified by the traffic matrix such that:

- the overall cost of the network is minimized,
- the resulting logical topology remains connected and is capable of handling the entire traffic demand in case of all possible single physical link failure scenarios, and
- a feasible RWA over underlying physical topology can be determined for the set of lightpaths created.

The cost we are trying to minimize, in this thesis, is the total number of lightpaths. Since each lightpath requires a transmitter and a receiver, minimizing the number of lightpaths is equivalent to minimizing the cost of transmitters and receivers. With current WDM technology, it is possible to have over a hundred of channels per fiber. Therefore the transceivers, rather than the wavelength channels, are becoming the scarce resources and it is important to minimizing their use.

3.4 Heuristic for Survivable Traffic Grooming

In the first step of the design process of our heuristic for survivable traffic grooming, (we call it heuristic *H-STG*), we use another heuristic (we call it heuristic *"create_topology")* to create an initial logical topology, which is capable of supporting the required traffic under fault-free conditions. Then, by considering each potential single link failure scenario, we augment this initial topology to create a robust logical topology that remains connected and can handle the entire traffic request in any single link failure scenario. We have given a detailed description of the heuristic *create_topology* in the next section. Many heuristics have been proposed for designing logical topologies [3], [15], which could be easily adapted for our purpose. Heuristic H-STG does not depend on the choice of a specific algorithm for logical topology design. We have designed our heuristic H-STG to follow the Generic Graph Model for Traffic Grooming proposed in [72], using the minimum lightpath (MinLP) policy with the Maximum-Amount-First (MAF) scheme implementing non-bifurcated traffic grooming. We describe the heuristic as below:

The heuristic H-STG first creates, using another heuristic called *create_topology,* an initial logical topology and a routing scheme capable of supporting the entire traffic request under fault-free conditions. Then it takes into account each potential single-link failure scenario, augments the logical topology whenever needed, and updates the routing scheme to take account of each failure scenario. In this process of augmentation, we use the same heuristic *create_topology* except that, now it has to take into account an existing logical topology that is already carrying some traffic and create a new logical topology, which handles the single-link failure scenario being considered.

Heuristic: H-STG (G_P, Q)
Inputs:

- The physical topology represented by the directed graph $G_P = (V_P, E_P)$
- A set of traffic request Q

Outputs:

- A set of lightpaths representing the logical topology LT capable to handle the entire traffic request in Q during the fault free situation and in case of any single link failure scenario.
- A set of routing information RT for all the requests in Q.

```
1.  begin
2.  |  (LT_init, RT_init) ← create_topology (G_P, ∅, ∅, Q).
3.  |  if (LT_init = ∅) then
4.  |  |  return (∅,∅).
5.  |  end if
6.  |  (LT, RT) ← (LT_init, RT_init).
7.  |  for each e ∈ E_P, do
8.  |  |  G_P^new ← (V_P, E_P \ {e}).
9.  |  |  LT_faulty ← set of logical edges in LT, which traverse link e.
10. |  |  LT_surv ← LT \ LT_faulty.
11. |  |  Q_new ← {q| q is a traffic request whose route RT^q over the logical
    |  |    topology involves an edge in LT_faulty}.
12. |  |  RT_surv ← RT \ {RT^q | q ∈ Q_new}.
13. |  |  (LT^e, RT^e) ← create_topology(G_P^new, LT_surv, RT_surv, Q_new).
14. |  |  if (LT^e = ∅) then
15. |  |  |  return (∅,∅).
16. |  |  else
17. |  |  |  LT ← LT^e ∪ LT_faulty.
18. |  |  |  RT ← RT_init.
19. |  |  end if
20. |  end for
21. |  return (LT, RT).
22. end
```

Figure 3.1 Overview of heuristic algorithm (H-STG)

In step 2 of our design process, we use *create_topology*, to generate an initial logical topology, LT_{init} and routing scheme, RT_{init}, for handling all traffic requests in Q, assuming

that the network is fault-free. At this point, since we currently do not have any existing logical topology, the initial topology and, as well as, the routing scheme for existing traffic, are both specified as empty lists. In other words, the second and the third input parameters of *create_topology*, are both empty lists. If we can create the initial topology successfully (i.e. $LT_{init} \neq \varnothing$, in step 3), the design process continues, otherwise the algorithm stops and reports failure.

In step 6, we start our design process using LT_{init} and RT_{init} and the logical topology LT and the routing scheme RT. Steps 8-19 are repeated for all physical links of the network. In a given iteration, we consider the case of a specific link e becoming faulty. In step 8 we update the physical topology by removing the faulty link e, giving the physical topology G_P^{new}. In step 9, we construct the set LT_{faulty} of lightpaths that are disrupted due to the failure of link e, by including in LT_{faulty} those lightpaths that traverse link e. In step 10, we create a temporary logical topology LT_{surv} that has survived the failure of link e. In step 11, we create a new traffic request matrix Q_{new} by determining the set of disrupted traffic requests that were using a logical path involving link e. Each request in Q_{new} has to be rerouted using an alternate logical path that does not use link e. The remaining requests are not affected by the failure of link e, and can use their existing routes. In step 12, we determine the routing scheme RT_{surv} for the surviving requests, by simply removing the disrupted requests from the original (fault-free) routing scheme.

In step 13, we try to reroute the affected traffic requests over the surviving logical topology, or determine which new logical edges, if any, have to be added to take care of

the requests. To do this, we invoke *create_topology* with the modified physical topology G_P^{new}, the surviving logical topology LT_{surv}, the routing scheme RT_{surv} for the surviving requests, and the traffic requirements Q_{new} representing the traffic originally carried by the link e. If all the disrupted traffic cannot be handled successfully by either utilizing the spare capacities in LT_{surv} or adding new lightpaths, then *create_topology* returns $LT^e = \emptyset$, and the algorithm stops, reporting failure. Otherwise *create_topology* returns a pair (LT^e, RT^e), where LT^e denotes an updated logical topology, including any new lightpaths added to accommodate the disrupted requests in Q_{new}, and RT^e denotes the routing scheme to be used when link e fails. We store the new routing scheme RT^e for future use whenever link e fails. This concludes iteration one of steps 8 – 19 - the process of considering a link e as faulty. When considering the case of failure of another link, since we are considering the case of single link failures only, we have to first restore the link e to a fault-free state. This means that the lightpaths that were disrupted when link e fails are now operational and must be included in the logical topology for the next iteration (step 17). When the lightpaths in LT_{faulty} are available, the routes specified in RT_{init} might all be used again, including the routes for the requests in Q_{new}. We do this in step 18. It should be noted that logical topology LT now includes logical edges added in step 13 to handle faulty link e. The next link failure scenario is then considered in the next iteration of steps 8 – 19 with the updated topology LT and the initial routing scheme RT_{init}.

3.4.1 Creating Logical Topology

Given an existing logical topology, already supporting a number of requests for data communication, the objective of the heuristic *create_topology* is to route, if possible, all

the requests in a set of additional requests for data communication. For this purpose, the residual capacity of the logical topology is utilized to the maximum extent possible. If the residual capacity is not sufficient to handle some of the requests, the heuristic augments the initial logical topology by adding a minimum number of additional logical edges. When the heuristic succeeds, it returns a new logical topology that can support all the requests for data communication previously supported by the initial logical topology as well as the specified set of additional requests for data communication. The heuristic also returns a routing scheme for all the requests the logical topology supports. The heuristic *create_topology* handles each new traffic request in Q without disturbing the logical edges already allotted to the requests handled so far. The routing scheme returned by *create_topology* for handling all the requests in Q must ensure that the total traffic on any logical edge does not exceed the capacity of a lightpath.

A special case arises when we invoke the heuristic *create_topology* for the very first time in the design process where there is no initial logical topology. In this case, the problem is to generate a logical topology to support a specified set of requests for data communication. This is the classical logical topology design problem [3], [15], [49], [50] that has been studied extensively in the literature. A heuristic to create the initial topology, which is similar to our heuristic, was presented in [55] where bifurcated traffic grooming was considered, whereas we have considered non-bifurcated traffic grooming.

The heuristic *create_topology* has to take into account:

- The physical topology represented by the directed graph $G_P = (V_P, E_P)$

- An initial logical topology LT_{init} expressed as a list of logical edges, each specified by its source and destination nodes and the total traffic carried by the it.
- A list of routing information RT_{init} where the i^{th} element is a pair (P_i, q_i) of the logical path Pi over the logical topology LT_{init} to handle the i^{th} request q_i[3].
- A set of traffic requests Q, where each request is specified by its source, its destination and the data communication rate in OC-n notation.

If the heuristic terminates successfully, it generates:

- A new logical topology LT that can support all the requests for data communication in Q in addition to all the requests in RT_{init}.
- A routing scheme RT to handle all the requests mentioned above.

We describe the heuristic *create_topology* as follows:

At the beginning, in step 2, we create R link-disjoint shortest paths (R = 3, in our experiment) for each node pair $(i, j \mid i, j \in V_P)$. In step 3, we look at all triplets (s, d, q) in set Q and select the request, a triplet $(s_{max}, d_{max}, q_{max})$ having the largest entry, say, q_{max}.

In steps 5 – 22, we handle, if possible, this request $(s_{max}, d_{max}, q_{max})$. The steps 5 to 22 will be repeated as long as there is any traffic request left in the set Q, that is, the process will be executed for the entire traffic request in set Q, unless the process terminates due to some other reason (as in step 15).

[3] For the special case, when we invoke the heuristic *create_topology* for the very first time, we make both lists LT_{init} and RT_{init} empty.

Heuristics: *create_topology* (G_P, LT_{init}, RT_{init}, Q)

Input:

- The physical topology represented by the directed graph $G_P = (V_P, E_P)$
- An initial logical topology LT_{init}
- A list RT_{init} specifying how existing requests have been routed
- A set Q of additional traffic requests that need to be handled.

Output:

- A logical topology LT capable of handling all the traffic request in Q, in addition to the traffic requests originally carried by LT_{init}.
- A set of routing information RT for all the requests in Q, including those already in RT_{init} as well as those in Q.

```
1.  begin
2.  |  disjoint_routes ← find_disjoint_routes(R, G_P)
3.  |  (s_max, d_max, q_max) ← max {q | (s, d, q) ∈ Q}
4.  |  while (q_max > 0) do
5.  |  |  logicalpath ← find_lightpaths (s_max, d_max, q_max, LT_init)
6.  |  |  if (logicaltpath ≠ ∅) then
7.  |  |  |  (LT, RT) ← update_lightpath_route (s_max, d_max, q_max, LT_init, RT_init)
8.  |  |  |  Q ← {Q \ (s_max, d_max, q_max)} ∪ {(s_max, d_max, 0)}
9.  |  |  else
10. |  |  |  routeFound ← find_route (s_max, d_max, q_max, disjoint_routes)
11. |  |  |  if (routeFound = true) then
12. |  |  |  |  (LT, RT) ← update_lightpath_route (s_max, d_max, q_max, LT_init, RT_init)
13. |  |  |  |  Q ← {Q \ (s_max, d_max, q_max)} ∪ {(s_max, d_max, 0)}
14. |  |  |  else
15. |  |  |  |  return (∅, ∅)
16. |  |  |  end if
17. |  |  end if
18. |  |  (s_max, d_max, q_max) ← max {q | (s, d, q) ∈ Q}
19. |  |  if (q_max > 0) then
20. |  |  |  LT_init ← LT
21. |  |  |  RT_init ← RT
22. |  |  end if
23. |  end while
24. |  return (LT, RT)
25. end
```

Figure 3.2 Overview of algorithm *create_topology*

There are two ways to handle the request

i) by using existing lightpaths or

ii) by setting up new lightpath (s).

For the very first entry (overall highest traffic), there will be no existing lightpaths since this is the first request to handle. For the subsequent requests, we will have some existing lightpaths. In step 5 we search the current logical topology LT_{init} to check if the request q_{max} could be handled using the spare capacity of the existing lightpaths only. The function *find_lightpaths* returns the shortest logical path that can handle request $(s_{max}, d_{max}, q_{max})$, from its source s_{max} to its destination d_{max}. If such a path is found (i.e. $logicaltpath \neq \varnothing$, in step 6), we simply send the request $(s_{max}, d_{max}, q_{max})$ through that logical path. In step 7, we augment the logical topology LT_{init} into LT by reducing the spare capacity of the lightpaths on the path by q_{max}, the amount of data communication using (OC-n notation), and augment the routing scheme RT_{init} by adding the routing information for request $(s_{max}, d_{max}, q_{max})$, giving RT. In step 8 we set q_{max} to 0.

If no suitable logical path is found in step 5, then we try to establish a new lightpath for the request $(s_{max}, d_{max}, q_{max})$. In step 10, we try to find the best physical path among the R disjoint routes we have already found out in step 2. Using this physical path, we establish a new lightpath from source s_{max} to destination d_{max}. If such a physical path is found ($routeFound = true$, step 11), then we set up the lightpath. Since the new lightpath is carrying traffic q_{max}, we reduce the data carrying capacity of the lightpath by q_{max}. In step 12 we augment:

a) the logical topology LT_{init} by this new logical edge giving LT,

b) the routing scheme RT_{init} by adding the routing information of q_{max} giving RT

Since we have finished with the request, in step 13 we set q_{max} to 0. If, in case no suitable physical path is found in step 11, to establish a lightpath, then the algorithm stops and it returns $(\varnothing, \varnothing)$ indicating failure in step 15.

If the algorithm continues, then in step 18, we look for the next largest request (s_{max}, d_{max}, q_{max}) in Q. If there is still a request to be handled, then the new logical topology LT and the new routing scheme RT is set as LT_{init} and RT_{init} in steps 20 and 21 respectively, to be considered in the next iteration. If there is no request left in Q, then in step 24 the algorithm terminates successfully returning the latest logical topology LT and the latest routing scheme RT.

As we have explained in Chapter 4, we have compared the properties of the logical topologies designed using our heuristic H-STG with those designed using dedicated path protection and shared path protection schemes. To carry out the experiments, we have also developed heuristics to design survivable logical topologies using dedicated path protection and shared path protection. We have called those heuristics H-DP (for dedicated path protection), and H-SP (for shared path protection). Both the heuristics are functionally almost identical to the heuristic *create_topology*, except for some differences in the we implemented the function *"find_route"* in steps10. Instead of finding only one suitable route to establish a lightpath as done in H-STG, the function *find_route* has to

find two routes for both H-DP and H-SP - one to establish the primary lightpath and the second to establish the backup lightpath. Both the heuristics H-DP and H-SP follow the well-known features of dedicated path protection and shared path protection schemes respectively. We have provided details of the function *find_route* in the section 3.4.4.

3.4.2 Creating Link-disjoint Paths

At the beginning of our design process, before starting our heuristic *create_topology*, we have to do some housekeeping jobs on the input data provided. One of which is to find R link-disjoint shortest paths for each source-destination pair in the network. These link-disjoint paths are created by successively applying Dijkstra's algorithm [33] to each node pair. After finding a shortest path for a particular source-destination pair, we delete the links used by this shortest path from the physical topology. We then use this modified topology to find next shortest path using the same algorithm. This process is repeated until R shortest paths are found. The reason for these steps is that, when we create a new lightpath from a source node s to a destination node d, then we can use one of these R paths to route the lightpath. It is much more efficient to check the R pre-defined paths, than to search all possible paths in the physical network for a suitable shortest path. Using R distinct routes also offers more flexibility than only considering the shortest route. We first try to establish the lightpath through the shortest among the R disjoint paths. If we are unable to assign a wavelength to the lightpath on this path, then we use the next shortest path. This provides greater flexibility in routing and wavelength assignment for a particular source-destination pair and usually keeps the path lengths small. Therefore, using R link-disjoint routes is a reasonable trade-off between searching all possible paths

and using only the shortest path. It has been shown in [3] that, in general, making $R = 3$, is a good choice.

3.4.3 Sending Traffic Using Existing Logical Edges

Figure 3.3 shows our implementation of the function *find_lightpaths*, to find the shortest logical path (i.e., the logical path having the fewest number of logical edges) from the source s_q to the destination d_q so that the request $q \in Q$ can be handled, using the spare capacity of the logical edges in the current logical topology LT. This corresponds to step 5 of the algorithm *create*_topology as described in Figure 3.2. The algorithm *find_lightpaths* uses the breadth-first search algorithm. In any iteration, V_0 denotes the set of new nodes visited in the last iteration V_1 denotes the set of nodes that we have not visited yet, V_C denotes the set of new nodes that we visit in this iteration. In step 2, we initialize V_0 *with* a set consisting of node s_q, since we start our search with the node s_q. In step 3, we initialize V_C to be a null set since the searching has not yet been started. In step 4, we define V_1 as the set of all nodes not yet been visited, or have not become a member of V_C, so far. We initialize V_1 with all the nodes in V_P, except s_q. In steps 5 and 6, we define two flags *route-found* and *search*-failed, and initialize both of them as false. In step 7, we define a list named *logicalpath* to store the chosen logical path and initialize it as an empty list. In step 8, we indicate that the steps 9 to 18 will be repeated until any one of the flags is true. In step 9, V_C becomes the set of all nodes j that are the neighbors of node i in the set V_0 and there is a logical edge $i \rightarrow j$ with a spare capacity of more than or equal to q. If V_C is found to be empty in step 10, then the flag *search-failed* is set to *true*

in step 11, that ultimately terminates the *while* loop and the algorithm terminates with a failed search.

Algorithm: *find_lightpaths* (s_q, d_q, q, *LT*, *RT*)

Input:

- A traffic request, specified by its source s_q, destination d_q and amount of data q.
- The current logical topology *LT*

Output:

- A list of logical edges denoting a logical path from s_q to d_q, having the shortest possible length, if the algorithm terminates successfully; otherwise, an empty list.

```
1.  begin
2.  |   V0 ← {sq},
3.  |   VC ← Ø
4.  |   VI ← VP - V0,
5.  |   route-found ← false,
6.  |   search-failed ← false,
7.  |   logicalpath ← Ø.
8.  |   while ((route-found = false) and (search-failed = false)) do
9.  |   |   VC ← {j | j ∈ VI, j ∉ V0, and a node i ∈ V0 is connected to node j ∈ VI
    |   |   with a logical edge having a spare capacity of at least q}
10. |   |   if (VC = Ø) then
11. |   |   |   search-failed ← true
12. |   |   else if (dq ∈ VC) then
13. |   |   |   logicalpath ← list of logical edges of the shortest path from sq to dq
14. |   |   |   route-found ← true.
15. |   |   else
16. |   |   |   VI ← VI - VC
17. |   |   |   V0 ← VC
18. |   |   end if
19. |   end while
20. |   return (logicalpath)
21. end
```

Figure 3.3 Overview of function *find_lightpaths*

Otherwise, in step 12, we check if the destination node of q, d_q, is a member of the set V_C. If that is true, that indicates that we have found a logical path from s_q to d_q. We copy this path in the list *logicalpath* in step 13 and set the flag *route-found* to true in step 14, which ultimately terminates the *while* loop indicating a success[4]. If, on the other hand, d_q is not yet a member of the set V_C, in step 16 we augment the set V_I by removing the nodes that are now members of V_C. In step 17, we set V_0 equivalent to V_C for the next iteration and the *while* loop continues. At the end, in step 20, the algorithm either returns a logical path from s_q to d_q indicating success, or returns an empty list indicating failure.

3.4.4 Creating New Lightpaths

The function find_route in step 10 of Figure 3.2 creates a single lightpath from s_q to d_q, if possible, using the shortest possible physical route over the physical network that is currently available. In step 2 of the algorithm *create_topology* (shown in Figure 3.2), we have already created a list of *R-disjoint-routes* for every node-pair. Figure 3.4 shows the details of function *find_route*. From the list of disjoint-routes from s_q to d_q, we choose the route with a minimum number of links, such that the same channel is available on all links in the route. This ensures that the wavelength continuity constraint is satisfied. If no single channel is available on all links in the shortest route, then we go for the next shortest route, and so on. Once a route with a channel is chosen, we assume that the logical edge is established, and we update the logical topology by adding this logical edge. Since this logical edge will carry request q, its data carrying capacity will be reduced by the data communication rate of the request q.

[4] This is a standard breadth first search. To simplify the description we have omitted details of how we keep track of the entire path while we search for the shortest path.

Function: *find_route* (s_q, d_q, q, *disjoint_routes*)
Input:
- A traffic request specified by it source, destination and amount
- A set of R disjoint routes over the physical topology

Output:
true, if a route is found, *false,* otherwise.

```
1.  begin
2.  |  r ← 0
3.  |  Λ ← (set of all the channel in a link)
4.  |  while (r < R) do
5.  |  |  p ← r^th pre-computed shortest path from s_q to d_q
6.  |  |  if (p = ∅) then
7.  |  |  |  return (false)
8.  |  |  else
9.  |  |  |  for each link e in p do
10. |  |  |  |  Λ_e ← set of available channels on link e
11. |  |  |  |  Λ ← Λ ∩ Λ_e
12. |  |  |  end for
13. |  |  |  if (Λ ≠ ∅) then
14. |  |  |  |  wavelength ← any (λ | λ ∈ Λ)
15. |  |  |  |  Reserve wavelength on each fiber in route p
16. |  |  |  |  return (true)
17. |  |  |  else
18. |  |  |  |  r ← r + 1
19. |  |  |  end if
20. |  |  end if
21. |  end while
22. |  return (false)
23. end
```

Figure 3.4 Overview of function *find_routes*

In step 2, we set r to be *0* indicating that we will first consider the shortest route among R disjoint routes from s_q to d_q. In step 3, we initialize a set of available wavelength channel Λ with all the channels available in a link. Step 4 indicates that the steps 5 to 20 will be repeated for all the R disjoint routes, unless we find a suitable channel to establish a lightpath from s_q to d_q in step 16. In step 5, we set p to be the r^{th} disjoint route that we

will consider in this iteration. If we do not find such path as in step 6, the algorithm stops and returns *false* in step 8, indicating failure. Step 9 indicates that the steps 10 and 11 will be repeated for all the physical links e in the route p. In step 10, we create another set of wavelength channel Λ_e that will have all the available channel in link e. In step 11, we take the intersection of set Λ and Λ_e, and call the new set Λ. After considering all the links in the route p, if we find an available channel in the r^{th} route ($\Lambda \neq \varnothing$, in step 13), then in step 14,we choose any wavelength channel λ from the set of available channels in Λ. We then reserve the wavelength channel λ to establish the new lightpath from s_q to d_q over this r^{th} route p in step 15. The algorithm then returns *true* in step 16 indicating success. If, on the other, hand we find no available channel ($\Lambda = \varnothing$, in step 13), then the while loop goes for the next iteration after setting $r = r+1$ in step 18, which indicates that we will now consider the next shortest route. If the algorithm cannot find any usable channel in any route among the R disjoint routes, it returns *false* in step 22, indicating failure.

We have explained the function *find_route* above as it is implemented in the function *create_topology* in the heuristic H-STG that we have proposed. We have used this function with different implementations in both heuristic for dedicated path protection H-DP (as *find_routes_dedicated*) and heuristic for shared path protection S-SP (as *find_routes_shared*). We have presented both of those implementations in figures 3.5 and 3.6 below.

Function: *find_routes_dedicated (s_q, d_q, q, disjoint_routes)*
Input:
- A traffic request specified by it source, destination and amount
- A set of R disjoint routes over the physical topology

Output:
- *true*, if routes are found, *false*, otherwise.

```
1.  begin
2.  | r ← 0
3.  | Λ ← (set of all the channel in a link)
4.  | while (r < R) do
5.  | | p_p ← r^th pre-computed shortest path from s_q to d_q
6.  | | if (p_p = ∅) then return (false)
7.  | | else
8.  | | | for each link e in p_p do
9.  | | | | Λ_e ← set of available channels on link e
10. | | | | Λ ← Λ ∩ Λ_e
11. | | | end for
12. | | | if (Λ ≠ ∅) then
13. | | | | wavelength ← any (λ | λ ∈ Λ)
14. | | | | Reserve wavelength on each fiber in route p_p
15. | | | | Λ ← (set of all the channel in a link)
16. | | | | s ← r + 1
17. | | | | while (s < R) do
18. | | | | | p_b ← s^th pre-computed shortest path from s_q to d_q
19. | | | | | if (p_b = ∅) then return (false)
20. | | | | | else
21. | | | | | | for each link e in p_b do
22. | | | | | | | Λ_e ← set of available channels on link e
23. | | | | | | | Λ ← Λ ∩ Λ_e
24. | | | | | | end for
25. | | | | | | if (Λ ≠ ∅) then
26. | | | | | | | wavelength ← any (λ | λ ∈ Λ)
27. | | | | | | | Reserve wavelength on each fiber in route p_b
28. | | | | | | | return (true)
29. | | | | | | else s ← s + 1
30. | | | | | | end if
31. | | | | | end if
32. | | | | end while
33. | | | else r ← r + 1
34. | | | end if
35. | | end if
36. | end while
37. | return (false)
38. end
```

Figure 3.5 Overview of function *find_routes_dedicated*

Function: *find_routes_shared (s_q, d_q, q, disjoint_routes)*
Input:

- A traffic request specified by it source, destination and amount
- A set of *R* disjoint routes over the physical topology

Output:

- *true*, if routes are found, *false*, otherwise.

```
1.  begin
2.  | r ← 0
3.  | Λ ← (set of all the channel in a link)
4.  | while (r < R) do
5.  | | p_p ← r^th pre-computed shortest path from s_q to d_q
6.  | | if (p_p = ∅) then return (false)
7.  | | else
8.  | | | for each link e in p_p do
9.  | | | | Λ_e ← set of available channels on link e
10. | | | | Λ ← Λ ∩ Λ_e
11. | | | end for
12. | | | if (Λ ≠ ∅) then
13. | | | | wavelength ← any (λ | λ ∈ Λ)
14. | | | | Reserve wavelength on each fiber in route p_p
15. | | | | Λ ← (set of all the channel in a link)
16. | | | | s ← 0
17. | | | | while ((s < R) and (s ≠ r)) do
18. | | | | | p_b ← s^th pre-computed shortest path from s_q to d_q
19. | | | | | if (p_b = ∅) then return (false)
20. | | | | | else
21. | | | | | | for each link e in p_b do
22. | | | | | | | Λ_e ← set of available channels on link e(shared or otherwise)
23. | | | | | | | Λ ← Λ ∩ Λ_e
24. | | | | | | end for
25. | | | | | | if (Λ ≠ ∅) then
26. | | | | | | | wavelength ← (λ | λ ∈ Λ, λ is shared in maximum links)
27. | | | | | | | Reserve wavelength on each fiber in route p_b
28. | | | | | | | return (true)
29. | | | | | | else s ← s + 1
30. | | | | | | end if
31. | | | | | end if
32. | | | | end while
33. | | | else r ← r + 1
34. | | | end if
35. | | end if
36. | end while
37. | return (false)
38. end
```

Figure 3.6 Overview of function *find_routes_shared*

3.5 An Example With a 4-Nodes Network

In this section we will explain algorithm H-STG with the help of an example. Figure 3.7 shows the physical topology of a four-node network and Table 3.1 contains the traffic demand matrix that we wish to handle. The objective of our algorithm is to design a logical topology with a minimum number of lightpaths that can handle all the traffic requirements and can withstand any single link failure.

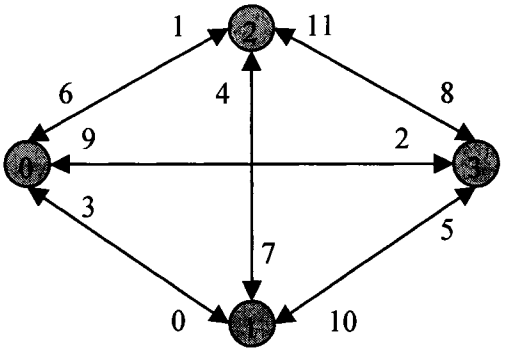


Figure 3.7 Physical topology

Nodes	0	1	2	3
0	0	OC-3x2 OC-12x2	OC-6x3 OC-24x1	OC-3x3 OC-12x1
1	0	0	OC-3x4 OC-24x1	0
2	0	OC-6x4 OC-12x2	0	0
3	OC-24x1	0	OC-12x1 OC-24x1	0

Table 3.1 Traffic Matrix, *T*

In the figure 3.7 for physical topology above, the number near each arrowhead denotes the distinct link number assigned to the link associated with the arrowhead. From the

above traffic matrix, we create a commodity set Q as in Table 3.2 below, where every commodity is a triplet corresponding to a request, consisting of the source, the destination and the volume of data communication using the OC-n notation. As we have already mentioned in section 3.4, our heuristic follows the Maximum-Amount-First (MAF) scheme [71] implementing non-bifurcated traffic grooming. To serve the purpose, we arrange the commodity set Q in such a way that, when we start processing the requests, we consider the commodity with the highest data communication rate first.

Commodity #	Source	Destination	Amount
0	0	2	OC-24
1	1	2	OC-24
2	3	0	OC-24
3	3	2	OC-24
4	0	1	OC-12
5	0	1	OC-12
6	0	3	OC-12
7	2	1	OC-12
8	2	1	OC-12
9	3	2	OC-12
10	0	2	OC-6
11	0	2	OC-6
12	0	2	OC-6
13	2	1	OC-6
14	2	1	OC-6
15	2	1	OC-6
16	2	1	OC-6
17	0	1	OC-3
18	0	1	OC-3
19	0	3	OC-3
20	0	3	OC-3
21	0	3	OC-3
22	1	2	OC-3
23	1	2	OC-3
24	1	2	OC-3
25	1	2	OC-3

Table 3.2 Commodity Set, Q

After completing step 1 of our heuristic H-STG (Figure 3.1), we get the initial logical topology (LT_0) as shown in the figure 3.8. The initial topology includes 6 lightpaths L_0, L_1, L_2, L_3, L_4 and L_5 (shown as dashed red lines) established over the physical topology.

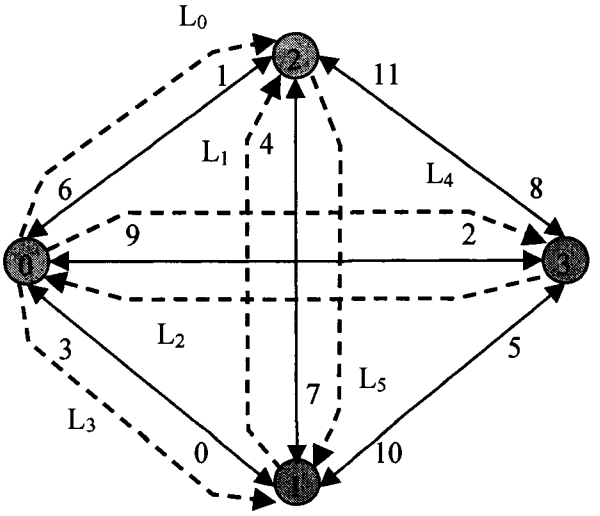


Figure 3.8 Logical topology before fault

Table 3.3 shows the RWA for each lightpath, including the link(s) used by the lightpath and the total traffic on the lightpath. Table 3.4 shows the detail routing information for each commodity, including the logical edges (lightpaths) used by each of them.

Lightpath	Link(s)	Total traffic on lightpath
L_0	$0 \rightarrow 2$	OC - 78
L_1	$1 \rightarrow 2$	OC - 36
L_2	$3 \rightarrow 0$	OC - 60
L_3	$0 \rightarrow 1$	OC - 30
L_4	$0 \rightarrow 3$	OC - 21
L_5	$2 \rightarrow 1$	OC - 48

Table 3.3 Information for each lightpath

Commodity #	Source	Destination (d_q)	Amount (q)	Lightpaths Used
0	0	2	OC-24	L_0
1	1	2	OC-24	L_1
2	3	0	OC-24	L_2
3	3	2	OC-24	L_2, L_0
4	0	1	OC-12	L_3
5	0	1	OC-12	L_3
6	0	3	OC-12	L_4
7	2	1	OC-12	L_5
8	2	1	OC-12	L_5
9	3	2	OC-12	L_2, L_0
10	0	2	OC-6	L_0
11	0	2	OC-6	L_0
12	0	2	OC-6	L_0
13	2	1	OC-6	L_5
14	2	1	OC-6	L_5
15	2	1	OC-6	L_5
16	2	1	OC-6	L_5
17	0	1	OC-3	L_3
18	0	1	OC-3	L_3
19	0	3	OC-3	L_4
20	0	3	OC-3	L_4
21	0	3	OC-3	L_4
22	1	2	OC-3	L_1
23	1	2	OC-3	L_1
24	1	2	OC-3	L_1
25	1	2	OC-3	L_1

Table 3.4 Routing information for each commodity

When our heuristic H-STG enters into the second phase, steps 7 – 20 are repeated to consider all single-link failure scenarios. Since the network in our example has 12 links, those steps have to be repeated 12 times. To illustrate the approach, we will investigate the case of a specific failure scenario in detail. The same process has to be repeated for all the physical links in the network.

Let us suppose that the fiber link 2 (0 → 3) fails. In step 9, we find the set of lightpath(s) using link 2. From figure 3.8, we can see that lightpath L_4 is affected by the failure of link 2. Now in step 11 of the heuristic, we find the commodities affected by the failure of lightpath L_4. From Table 3.4 we can find that commodities 6, 19, 20, and 21 uses lightpath L_4. Accordingly we create a new commodity set Q_{new} as Table 3.5 below:

Commodity #	Source	Destination	Amount
6	0	3	OC – 12
19	0	3	OC – 3
20	0	3	OC – 3
21	0	3	OC – 3

Table 3.5 Commodity set Q_{new} after fault in link 2

Now, to handle the traffic in Q_{new}, we call the *create_topology* function again in step 13. This function returns a new logical topology capable of handling the entire traffic request, not only in the fault-free condition, but also in the situation where the link 2 (from node 0 to node 3) becomes faulty. In this case, this is done by sending the affected traffic through the new logical edge L_6 that has been created during the process. Therefore, after only this iteration, the logical topology becomes as in the figure 3.9.

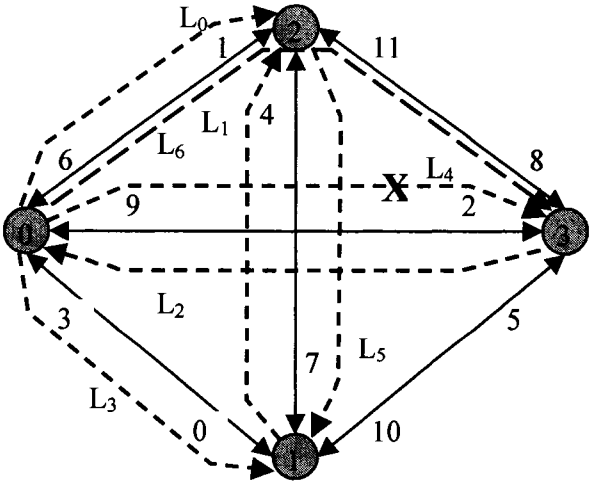


Figure 3.9 Logical topology after handling fault in link 2.

The routing scheme returned by this iteration is given below in Table 3.6. Whenever the link 2 fails, we simply re-route the entire traffic according to the given routing table. We can see from the table that we actually have to take care of only the requests in the shaded rows.

Commodity #	Source (s_q)	Destination (d_q)	Amount (q)	Lightpaths
0	0	2	OC-24	L_0
1	1	2	OC-24	L_1
2	3	0	OC-24	L_2
3	3	2	OC-24	L_2, L_0
4	0	1	OC-12	L_3
5	0	1	OC-12	L_3
6	0	3	OC-12	L_4
7	2	1	OC-12	L_5
8	2	1	OC-12	L_5
9	3	2	OC-12	L_2, L_0
10	0	2	OC-6	L_0
11	0	2	OC-6	L_0
12	0	2	OC-6	L_0
13	2	1	OC-6	L_5
14	2	1	OC-6	L_5
15	2	1	OC-6	L_5
16	2	1	OC-6	L_5
17	0	1	OC-3	L_3
18	0	1	OC-3	L_3
19	0	3	OC-3	L_4
20	0	3	OC-3	L_4
21	0	3	OC-3	L_4
22	1	2	OC-3	L_1
23	1	2	OC-3	L_1
24	1	2	OC-3	L_1
25	1	2	OC-3	L_1

Table 3.6 Routing information in case link 2 fails

To take care of all the link-failure scenarios, the heuristic repeats the whole process described above for every link in the network. The final topology returned by our heuristic H-STG, after considering every link-failure scenario, and adding the new lightpaths (L_6, L_7, L_8, and L_9) to the original initial logical topology, is shown in figure 3.10. This logical topology can handle the entire set of traffic requests, not only in the fault free condition, but also, in the case of *any* single link failure scenario.

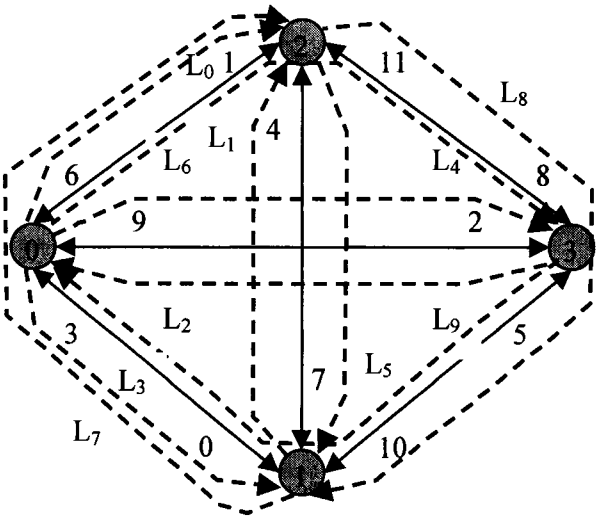


Figure 3.10 The final logical topology.

CHAPTER FOUR

4.0 EXPERIMENTS AND RESULTS

4.1 Overview

In this chapter, we present the results of the experiments for studying the performance of our approach for survivable traffic grooming, compared to some existing methods for designing robust survivable logical topologies. In particular, we compare the performances of topologies designed using our technique to those based on shared path protection (SP) and dedicated path protection (DP). We have carried out experiments on a number of networks of different sizes, ranging from 6 nodes to 40 nodes. For a given size of a network, we have randomly generated 5 distinct physical topologies, and we tested each topology using three categories of data communication rates. We have used traffic matrices as described in Chapter 2 to represent the data communication rates. We used three categories of data communication rates as follows:

- Low traffic: where the traffic is randomly distributed between 0 – OC-24.
- Medium traffic: where the traffic is randomly distributed between 0 – OC-48.
- High traffic: where the traffic is randomly distributed between 0 – OC-96.

We randomly generated 5 traffic matrices randomly for every category. Traffic requests between any node-pair may be either zero or any combination of OC-3, OC-6, OC-12 or OC-24, provided they are within the range give above. We have used the same traffic matrices to design logical topologies, using shared path protection, dedicated path

protection and our approach. We have used the value of $K = 64$ for all of our experiment, where K is the number of available channels per fiber. In our experiments we have also assumed that the numbers of transmitters and receivers available in each node is unlimited. We have made this assumption just to keep our algorithm simpler and enable us to focus on reducing the number of lightpaths and the number of channels. In practice, the numbers of available transceivers in any node are always limited. Extending the heuristics to include this restriction is straightforward.

The topologies generated by all three approaches are capable of surviving single link failures. Therefore, in order to evaluate our technique, we compare the resources required to implement such topologies. When considering resource requirements, for a given size of network with a specified category for the data communication rates, we are primarily interested in the following metrics:

- the average number of wavelength channels used per fiber, and
- the average number of lightpaths created.

In the introduction (Section 1.2) we have discussed why these metrics are important.

4.2 Results of Experiments on Channel Used

Table 4.1 presents the results of our experiments for the average number of channels required per fiber, to implement a survivable logical topology, using three heuristics as follows:

- H-STG implementing our proposed scheme for robust logical topology (section 3.4),
- H-SP implementing the shared path protection (section 3.4.1),
- H-DP implementing the dedicated path protection (section 3.4.1),

The entries represent the average values for all *successful* experiments, where the heuristic could generate a robust logical topology. If, for a particular size of the network and for a particular traffic category (i.e. high, medium or low) *all* the experiments failed then we have put a '–' for the corresponding entry. For example, for $N = 40$ there were no successful designs for "high" traffic loads for H-DP and H-SP and are shown as '-'. The actual traffic matrices used in the experiments are given in Appendix B.

Average number of channels used per fiber									
N	High Traffic			Medium Traffic			Low Traffic		
	H-DP	H-SP	H-STG	H-DP	H-SP	H-STG	H-DP	H-SP	H-STG
6	3.80	2.67	2.43	2.25	1.68	1.70	2.08	1.56	1.57
10	9.23	6.05	5.41	5.65	3.84	3.52	3.77	2.72	2.55
14	16.01	10.12	9.26	10.15	6.70	5.87	6.20	4.18	3.87
20	29.61	17.85	15.74	18.57	11.51	9.96	11.14	7.05	6.28
30	-	31.22	28.59	32.27	21.09	17.94	22.20	13.58	11.20
40	-	-	40.42	-	30.44	26.08	34.61	22.09	17.56

Table 4.1 Comparison of the average number of channels per fiber required

4.2.1 Performance Analysis on Channels Used

Figures 4.1, 4.2 and 4.3 show how the average numbers of channels vary with the sizes of the networks, for each of the heuristics used under different categories of traffic matrices. Figures 4.1 (respectively 4.2 and 4.3) depict the results corresponding to low (respectively medium and high) traffic loads. From the charts and the graphs, it is evident that both shared path protection and our heuristic clearly outperform dedicated path protection, in all cases. Our approach also performs better than shared path protection when the number of nodes in the network exceeds 6.

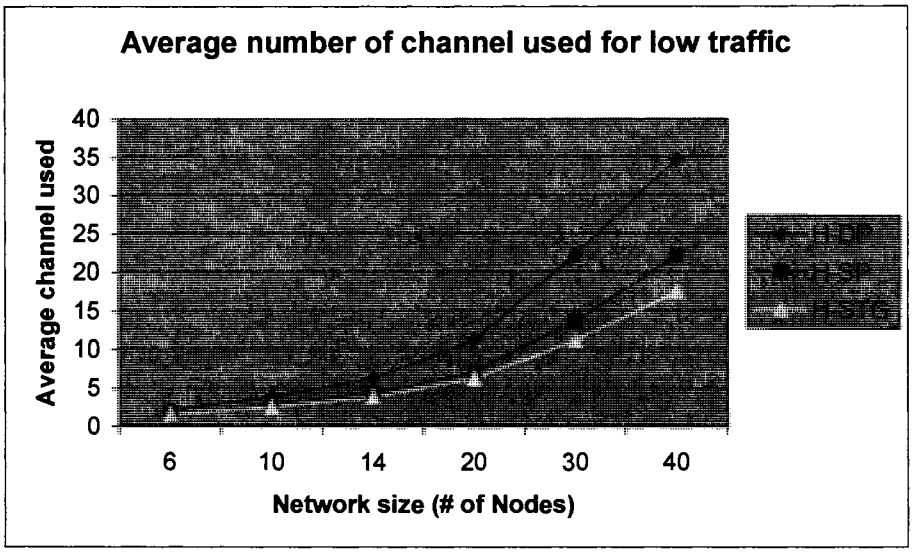


Figure 4.1 Avg. # of channels used per fiber for low traffic

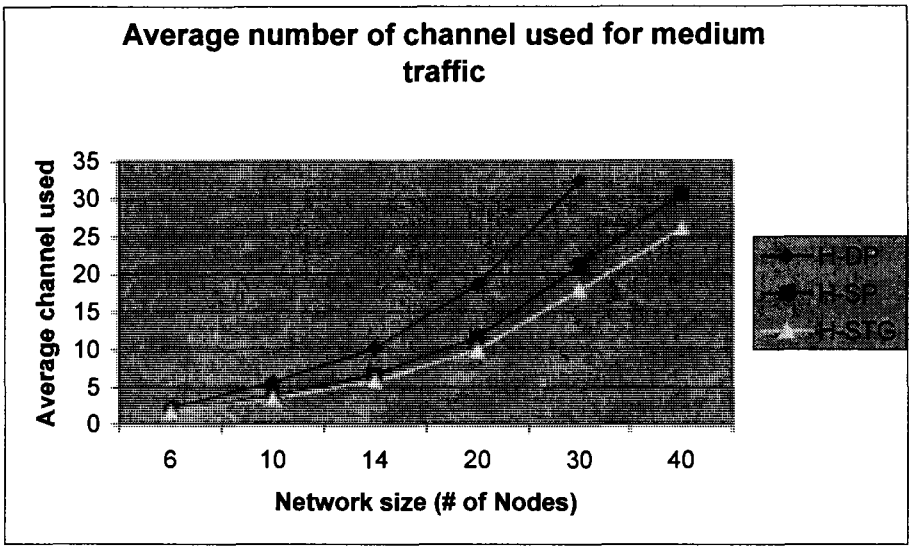


Figure 4.2 Avg. # of channels used per fiber for medium traffic

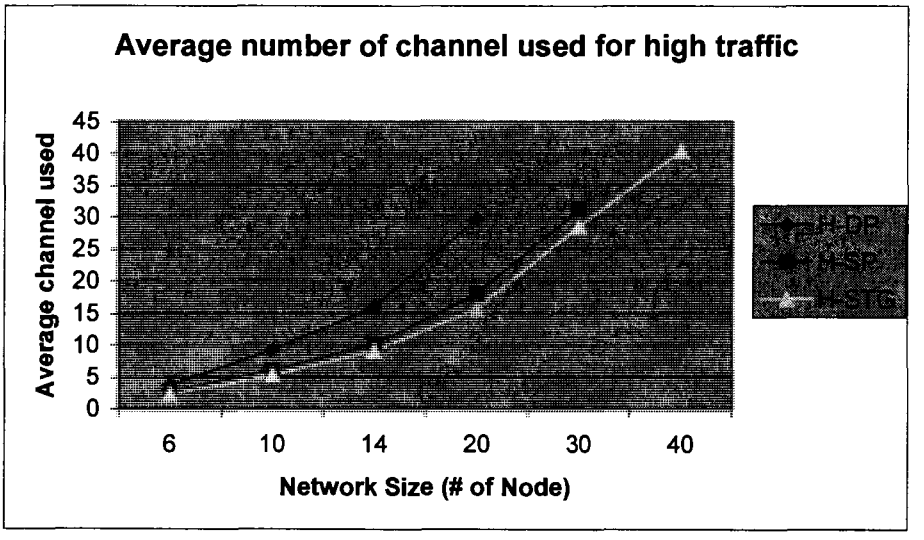


Figure 4.3 Avg. # of channels used per fiber for High traffic

4.3 Results of Experiments on The Number of Lightpaths

In addition to a comparison of the number of channels used per fiber, more importantly in our experiments, we have also compared the number of lightpaths needed to implement a logical topology. We have already mentioned in Section 1.2 that a lightpath requires one transmitter and one receiver. Therefore, it is important to try to minimize the number of lightpaths and hence the cost of transmitters and receivers. Table 4.2 compares the average number of lightpaths required to generate feasible topologies for networks of different sizes and with different categories of traffic matrices. As before, when calculating averages, we only consider the experiments where a topology could be successfully designed to handle the required traffic.

Average number of lightpaths created									
N	High Traffic			Medium Traffic			Low Traffic		
	H-DP	H-SP	H-STG	H-DP	H-SP	H-STG	H-DP	H-SP	H-STG
6	38	38	27	22	22	18	20	20	16
10	110	110	78	68	68	50	46	46	35
14	220	220	155	140	140	99	86	86	65
20	490	490	320	306	306	204	184	184	127
30	-	1174	733	746	746	464	468	468	291
40	-	-	1290	-	1370	810	932	932	543

Table 4.2 Comparison of the average number of lightpaths required

4.3.1 Performance Analysis on the Number of Lightpaths Used

Figures 4.4, 4.5, and 4.6 represent the average number lightpaths required in the successful experiments, for networks of different sizes, under different load conditions.

For a given logical topology, both dedicated path protection and shared path protection generates networks requiring the same number of lightpaths. Table 4.2 also supports this fact. Therefore to compare the number of lightpaths generated and to calculate the increase in performance, we have only used the data generated by our heuristic H-STG and the data generated by the heuristic for shared path protection H-SP. Figures 4.4, 4.5 and 4.6 show that the number of lightpaths required in our approach is consistently less than the total number of lightpaths required by shared/dedicated path protection scheme. We can also see that, with the increase of the size of the network, the improvement becomes more and more significant.

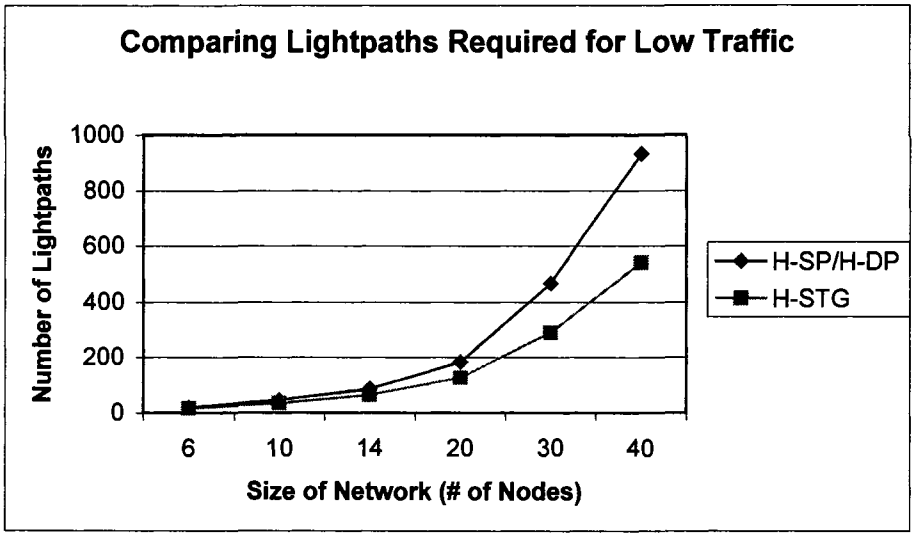


Figure 4.4 Avg. # of lightpaths used for low traffic

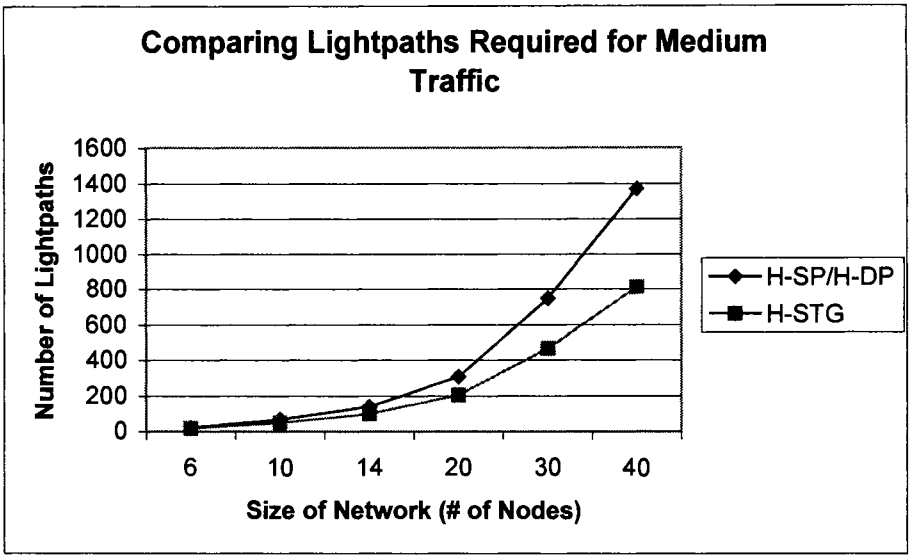


Figure 4.5 Avg. # of lightpaths used for medium traffic

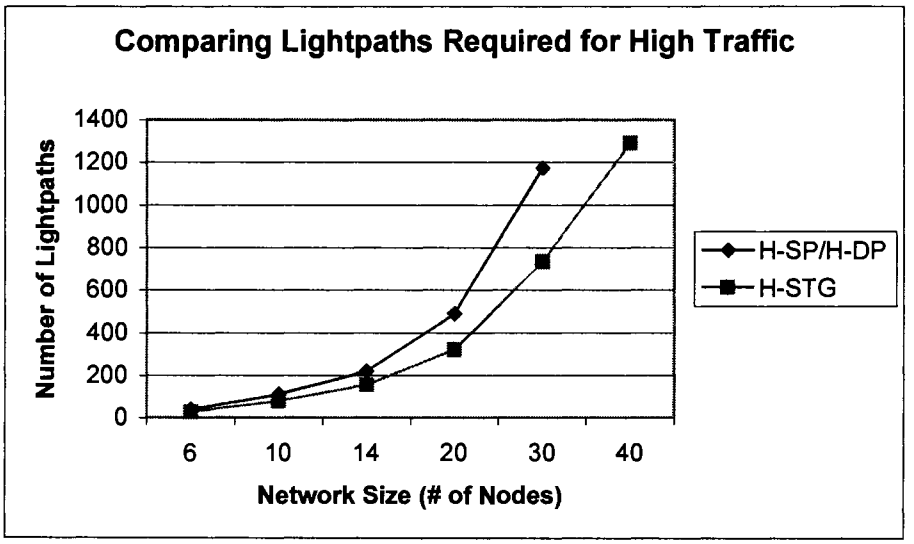


Figure 4.6 Avg. # of lightpaths used for high traffic

4.4 Performance Improvement in Our Approach

In this section, we summarize the percentage improvement of dedicated protection (H-DP vs. H-STG) and shared protection versus our approach (H-SP vs. H-STG), in terms of the number of channels used. We have also summarized the percentage of improvement for the number of lightpaths needed to design logical topology using the three approaches.

Since both the dedicated path protection and the shared path protection generate the same number of lightpaths for a specific logical topology, we have done the comparison with our approach to only one path protection approach, the shared path protection.

Table 4.3 shows the values of percentage improvement in the number of channels used. If there is no valid entry for a cell in the table, we have put a '-' in the corresponding cell in Table 4.3.

Percentage improvement in number of channels used per fiber						
N	High Traffic		Medium Traffic		Low Traffic	
	H-DP Vs. H-STG	H-SP Vs. H-STG	H-DP Vs. H-STG	H-SP Vs. H-STG	H-DP Vs. H-STG	H-SP Vs. H-STG
6	36.05	8.99	24.44	-1.19	24.52	-0.64
10	41.39	10.58	37.70	8.33	32.36	6.25
14	42.16	8.50	42.17	12.39	37.58	7.42
20	46.84	11.82	46.37	13.47	43.63	10.92
30	-	8.42	44.41	14.94	49.55	17.53
40	-	-	-	14.32	49.26	20.51

Table 4.3 Avg. % improvements in channel usage per fiber

Figure 4.7 depicts the average improvement in channel usage, for different load conditions - high, medium and low. The graph clearly shows how the improvement increases with the size of the network. For example, in the case where we compare the shared path protection to our approach, the improvement is, on an average, about 2% for 6-node networks to almost 18% for 40-node networks. This improvement is much higher in case of the dedicated path protection scheme when compared to our approach. The charts resulting from the experiments are given in Appendix E.

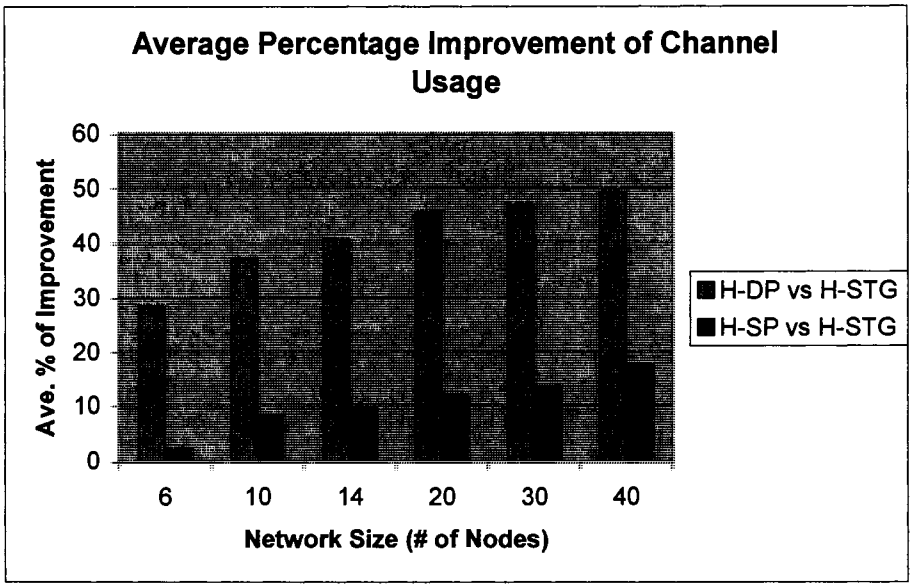


Figure 4.7 Avg. % improvements in channel usage per fiber

Table 4.4 shows the percentage improvement in the number of lightpaths created, using our approach over the shared path protection, under high, medium and low traffic categories. The values are calculated in exactly the same manner as in Table 4.3.

Figure 4.8 depicts the improvement in the number of lightpaths used, averaged over different traffic matrix categories. The graph shows an encouraging improvement in the performance regarding the number of lightpaths needed, using our heuristic H-STG, as compared to both the shared path heuristic (H-SP) and the dedicated path protection heuristic (H-DP). It also clearly shows how the percentage improvement increases with size of the network, which is almost 22% for 6-node network to almost 42% for 40-node network.

Percentage improvement in number of lightpaths created			
N	High Traffic	Medium Traffic	Low Traffic
	H-DP/H-SP Vs. H-STG	H-DP/H-SP Vs. H-STG	H-DP/H-SP Vs. H-STG
6	28.95	18.18	20.00
10	29.09	26.47	23.91
14	29.55	29.29	24.42
20	34.69	33.33	30.98
30	37.56	37.80	37.82
40	-	40.88	41.74

Table 4.4 Avg. % improvements in lightpath usage

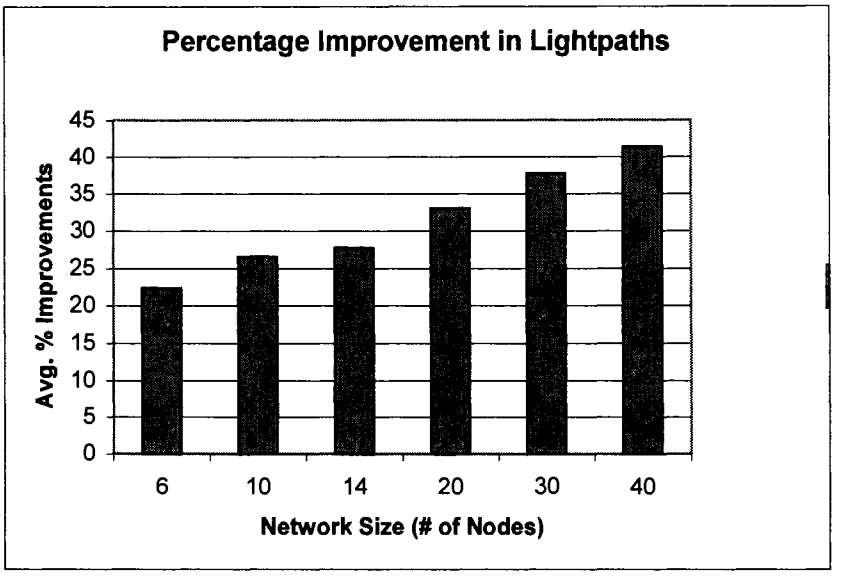


Figure 4.8 Avg. % improvements in lightpath usage

4.5 Statistical Analysis of The Experimental Results.

In this section, we analyze the statistical significance of our results. For each set of experiments, we calculate the 95% confidence interval (C.I). We have specified this interval by an upper bound (U) and a lower bound (L), and we are confident that in the 95% of the cases the mean of the samples will be with in the confidence limits L and U. The confidence interval is calculated using equation (4.1).

$$95\%\ \text{C.I.} = \bar{x} \pm 1.96 \times \frac{s}{\sqrt{n}} \quad \text{..... (4.1)}$$

Where, $\bar{x}$ = mean of the samples,

n = size of the samples, and

s = Standard Deviation, calculated using the equation 4.2

$$s = \sqrt{\frac{1}{n-1}\sum_{i-1}^{n}(x_i - \bar{x})^2} \quad \text{..... (4.2)}$$

4.5.1 95% Confidence Interval for Channels Used

Table 4.5 shows the values of 95% Confidence Interval for improvement in channel used for dedicated path protection vs. our approach and shared path protection vs. our approach, for different size of networks in different load conditions. For example, in Table 4.5, for a 10-node network, with low traffic conditions, the 95% C.I. is between 27.85 and 36.87, for H-DP vs. H-STG. This means that the mean of percentage improvement (in terms of the number of channels used per fiber) using dedicated path protection vs. our approach will not be less than 27.85 and will not be more than 36.87, 95% of the time.

95% Confidence Interval for % Improvement in Channels Use							
Nodes	Traffic Category	H-DP vs. H-STG			H-SP vs. H-STG		
		Lower Bound	Average Value	Upper Bound	Lower Bound	Average Value	Upper Bound
10	Low	27.85	32.36	36.87	1.82	6.25	10.68
	Medium	31.79	37.70	43.61	2.00	8.33	14.66
	High	34.87	41.39	47.91	2.93	10.59	18.25
20	Low	41.96	43.63	45.30	7.65	10.92	14.19
	Medium	44.42	46.37	48.33	9.52	13.47	17.42
	High	44.57	46.84	49.11	7.66	11.82	15.98
30	Low	40.63	43.63	46.63	16.06	17.53	19.00
	Medium	35.95	46.37	56.80	12.65	14.94	17.24
	High	-	-	-	5.15	8.42	11.69

Table 4.5 95% Confidence Interval for channel usage per fiber

4.5.2 95% Confidence Interval for the number of Lightpaths Used

Table 4.6 shows the values of 95% Confidence Interval for lightpaths used under different traffic load conditions, comparing dedicated path protection (or shared path protection) vs. our approach of survivable routing. For example, for 20 node networks and for medium traffic condition, 95% Confidence Intervals for H-DP/H-SP vs. H-STG is 26.44 and 33.33. In other words, the mean of percentage improvement of lightpaths used in dedicated path protection or shared path protection vs. our approach will not be less than 26.44 and will not be more than 33.33, 85% of the time.

95% Confidence Interval for % Improvement in Lightpath				
Nodes	Traffic Category	H-DP/H-SP vs. H-STG		
		Lower Bound	Average Value	Upper Bound
10	Low	14.34	23.91	33.48
	Medium	12.88	26.47	40.07
	High	15.29	29.09	42.89
20	Low	24.69	30.98	37.27
	Medium	26.44	33.33	40.22
	High	27.64	34.69	41.74
30	Low	36.95	37.82	38.69
	Medium	35.78	37.80	39.82
	High	34.43	37.56	40.69

Table 4.6 95% Confidence Interval for lightpath usage

CHAPTER FIVE

5.0 CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

The objective of this thesis was to develop an efficient technique for designing fault tolerant logical topologies for WDM networks. The final logical topology should be able to withstand any single link failure in the network. The input to the design process consisted of the underlying physical topology, the requests for data communication to be handled by the network (specified, for each situation, using a traffic matrix) and the resource constraints on the number of transmitters and receivers at each node. An ILP formulation for this problem could be used to generate optimal solutions. But, the problem becomes computationally intractable, even for relatively small networks. Therefore, an efficient heuristic was required to solve the problem.

In this thesis, we have presented a quick and efficient heuristic for fault tolerant logical topology design. One of the main objectives of our design process was to keep the cost of the network as low as possible. In current WDM networks, it is possible to support hundreds of WDM channels on a single fiber. Therefore, the cost of the transmitters and the receivers, and hence the number of lightpaths, is becoming the main factor in determining the cost of a WDM network. We have tried to keep the number of lightpaths required to implement a topology as low as possible. When we have processed a specific traffic request, we have tried to use existing lightpaths as much as possible.

We have tested our heuristics on number different sizes of the networks, ranging from 6 nodes to 40 nodes. For a given size of the network, we have carried out experiments with 75 data sets. We have compared our results with two widely used techniques for survivable network design - dedicated path protection and shared path protection techniques. The results clearly show that our heuristic provides a significant improvement over both the existing techniques, in terms of the number of WDM channels used as well as the number of lightpaths required to implement a topology.

It is important to note that, in many cases, we are able to design a feasible topology using our new heuristic H-STG, when both shared path and dedicated path protection schemes fail, for a given amount of available resources. Our heuristic is easily scalable and can be used for large WDM networks.

5.2 Future Works

While designing our heuristic H-STG, we have only considered the single link failures, since they are the predominant form of failures in optical networks [52]. However, our heuristic can easily be enhanced to consider the multiple link failure scenarios. At its present form, our heuristic consider each link separately, identifies the lightpaths in the link, determines the traffic flowing through them and tries to reroute those traffic through the surviving logical topology or creates new lightpaths if necessary. It is easily possible to modify our heuristic so that it can perform the entire process mentioned, considering two or more links at a time. We are willing to work on this problem at a near future time.

In our heuristic H-STG, while considering link failure scenarios, we have considered physical edges simply in the order of edge # 0, 1, 2...etc. It might be possible to optimize our heuristic by carefully considering the sequence in which we should consider the physical edges. Also, our heuristic specifies a routing strategy to route the traffic over the generated logical topology. Although this routing is feasible, it is not guaranteed to be optimal. Our heuristic can be enhanced, by adding another step, which takes the logical topology and traffic matrix, and optimally routes the traffic over the topology. Both of these jobs are significant works and are eligible for separate projects. We look forward to work on these areas at any future time.

BIBLIOGRAPHY

1. Arakawa, S., Murata, M., and Miyahara, H.; "Design methods of multiplayer survivability in IP over WDM networks", *In Proceeding OptiComm. TX.* October 2000.

2. Armitage, J., Crochat, O., and Le Boudec, J-Y.; "Design of a survivable WDM photonic network", *IEEE INFOCOM Proceedings of Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, Volume 1, Page(s): 244 – 252, April 1997.

3. Bandyopadhyay, S.; "Dissemination of Information in Optical Networks: From Technology to Algorithm", *Springer Publications, ISBN: 978-3-540-72874-0,* 2008.

4. Battiti, R. and Brunato, M.; "Reactive search for traffic grooming in WDM networks", *In Evolutionary Trends of the Internet, Proceedings of 2001, Tyrrhenian International workshop on Digital Communications,* Italy, Springer-Verlag, Page(s): 56 – 66, September 2001.

5. Boom, V. D. H. P. A., Li, W.; Bennekom, V. P. K., Monroy, I.T., and Khoe, G-D.: "High-capacity transmission over polymer optical fiber", *IEEE Journal of Selected Topics in Quantum Electronics*, Volume 7, Issue 3, Page(s):461 – 470, May – June 2001.

6. Brackett, C.; "Dense Wavelength Division Multiplexing Networks: Principles and applications", *IEEE Journal of Selected Areas in Communications*, Page(s): 373 – 380, August 1990.

7. Caenegem, B. V., Parys, W. V., Turck, F. D., and Demeester, P. M.; "Dimensioning of survivable WDM networks," *IEEE Journal on Selected Areas in Communications*, Volume 16, Page 1146 – 1157, September 1998.

8. Chiu, A. L. and Modiano, E.; "Reducing electronic multiplexing costs in unidirectional SONET/WDM ring networks via efficient traffic grooming", *Global Telecommunications Conference, GLOBECOM 98. The Bridge to Global Integration. IEEE,* Volume 1, Page(s): 322 – 327, November 1998.

9. Chlamtac, I. ; Ganz, A.; and Karmi, G.; "Lightpath communications: An approach to high bandwidth optical WAN's", *IEEE Transactions on Communications*, Volume 40, Page(s): 1171 – 1182, July 1992.

10. Chow, W. W. and Koch, S. W.; "Semiconductor-Laser Fundamentals: Physics of the Gain Materials", *Springer, ISBN 3540641661*, 1999.

11. Cowie, G. J., Yu, D., and Chieng Y. T.; "Brillouin/erbium fiber lasers", *Journal of Lightwave Technology,* Volume 15, Issue 7, Page(s): 1198 – 1204, July 1997.

12. Crochat, O., and Boudec, J-Y.L.; "Design Protection for WDM Optical Networks", *IEEE Journal on Selected Areas in Communications,* Volume 16, Issue 7, Page(s): 1158-1165, September 1998.

13. Doshi, B. T., Dravida, S., Harshavardhana, P., Hauser, O., and Wang, Y.; "Optical network design and restoration", *Bell Labs Technical Journal,* Page 58 – 83. January – March 1999.

14. Doucette, J., and Grover, W.D.; "Influence of modularity and economy-of-scale effects on design of mesh-restoration DWDM networks", *IEEE Journal on Selected Areas in Communications*, Volume 18, Page 1912 – 1923, October 2000.

15. Dutta, R., and Rouskas, G. N.; "A Survey of Virtual Topology Design Algorithms for Wavelength Routed Optical Networks," *Optical Network Magazine*, Volume 1, Issue 1, Page(s): 73 – 89, January 2000.

16. Dutta, R., and Rouskas, G. N.; "On optimal traffic grooming in WDM rings". *IEEE Journal on Selected Areas in Communications*, Volume 20, Issue 1, Page(s): 110 – 121, January 2002.

17. Dutta, R., Huang, S., and Rouskas, G. N.; "On optimal traffic grooming in elemental network topologies". *In Opticomm*, Page(s): 13 – 24, October 2003.

18. Ellinas, G., Hailemariam, A. G. and Stern, T. E.; "Protection cycles in mesh WDM networks," *IEEE Journal on Selected Areas in Communications*, Volume 18, issue 10, Page(s): 1924 – 1937, October 2000.

19. Fang, J., and Somani, A. K.; "Enabling Subwavelength Level Traffic Grooming in Survivable WDM Optical Network Design," *IEEE GLOBECOM Global Telecommunication Conference 2003*, Volume 1, Page(s): 2761 – 2766, 2003.

20. Frederick, M. T. and Somani, A. K.; "A single-fault recovery strategy for optical networks using subgraph routing". *Proceedings of the 7th Conference on Optical Network Design and Modeling (ONDM).* Page(s): 549 – 568, 2003.

21. Fumagalli, A., Cerutti, I., Tacca, M., Masetti, F., Jagannathan, R., and Alagar, S.; "Survivable networks based on optimal routing and WDM self healing rings", *Proc. of IEEE INFOCOM '99.* Volume 2, Page 726 – 733, March 1999.

22. Georges, J. B., Cutrer, D. M., Solgaard, O., and Lau, K.Y.; "Optical transmission of narrowband millimeter-wave signals", *Microwave Theory and Techniques, IEEE Transactions on,* Volume 43, Issue 9, Part 1–2, September 1995 Page(s): 2229 – 2240

23. Gerstel, O., and Ramaswami, R.; "Optical layer survivability-an implementation perspective," *IEEE Journal on Selected Areas in Communications*, Volume 18, Issue 10, Page(s): 1885 – 1899, October 2000.

24. Haque, A., Aneja, Y. P., Bandyopadhyay, S., Jaekel, A., and Sengupta, A.; "Some studies on the logical topology design of large multi-hop optical networks", *Optical Networks Magazine*, Volume 3, Issue 4, July/August 2002.

25. Ho, P. H. and Mouftah, H.T.; "Shared protection in mesh WDM networks," *IEEE Communications Magazine*, Volume 42, Issue 1, Page(s): 70 – 76, 2004.

26. Hu, J. Q. and Leida, B.; "Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks". *In IEEE INFOCOM*, Volume 1, Page(s) 495 – 501, March 2004.

27. Huang, S. and Dutta, R.; "Research Problems in Dynamic Traffic Grooming in Optical Networks", *Workshop on Traffic Grooming, IEEE Broadnets*, October 2005.

28. Huo, W., Guang, L., Assi, C. and Shami, A.; "Survivable traffic grooming in optical networks with multiple failures", *Electrical and Computer Engineering, 2005. Canadian Conference on*, Page(s): 1132 – 1135, May 2005.

29. Islam, M. N.; "Raman amplifiers for telecommunications", *Selected Topics Quantum Electronics, IEEE Journal of*, Volume 8, Issue 3, Page(s): 548 – 559, May – June 2002

30. Jaekel, A., Bari, A. and Bandyopadhyay, S.; "New techniques for efficient traffic grooming in WDM mesh networks". *In IEEE ICCCN, Optical networking Track*, 2007.

31. Joergensen, C., Durhuus, T., Braagaard, C., Mikkelsen, B. and Stubkjaer, K.E.; "4 Gb/s optical wavelength conversion using semiconductor optical amplifiers", *IEEE Photo. Tech. Lett.*, Volume 5, Issue 6, Page(s): 657 – 670, June 1993.

32. Konda, V. R. and Chow, T. Y.; "Algorithm for traffic grooming in optical networks to minimize the number of transceivers", *High Performance Switching and Routing, 2001 IEEE Workshop on,* Page(s): 218 – 221, May 2001.

33. Krishnaswamy, R. M. and Sivarajan, K. N.; "Design of logical topologies: a linear formulation for wavelength routed optical networks with no wavelength changers", *Proc. IEEE INFOCOM*, pages 919 – 927, 1998.

34. Lee, C. and Park, E. K.; "A genetic algorithm for traffic grooming in all-optical mesh networks", *IEEE International Conference on Systems, Man and Cybernetics, 2002,* Volume 7, Page(s): 6, October 2002.

35. Limal, E., Danielsen, S. L. and Stubkjaer, K .E.; "Capacity utilization in resilient wavelength-routed optical networks using link restoration", Proc., *OFC '998, San Jose,* CA, Volume 2, Page 297 – 298, February 1998.

36. Mahalati, R. and Dutta, R.; "Reconfiguration of traffic grooming optical networks", *Broadband Networks, 2004. BroadNets 2004. Proceedings. First International Conference on,* Page(s): 170 – 179, 2004.

37. Modiano, E. and Narula-Tam, A.; "Survivable routing of logical topologies in WDM networks", *INFOCOM 2001 Proceedings, IEEE*, Volume 1, Page(s): 348-357, 2001.

38. Mukherjee, B., Ramamurthy, S., Banerjee, D. and Mukherjee, A.; "Some principles for designing a wide-area optical network", *IEEE/ACM Trans. Networking*, Volume 4, Issue 5, Page(s): 684 – 696, October 1996.

39. Mukherjee, B.; "WDM optical communication networks: progress and challenges," *IEEE Journal on Selected Areas in Communications,* Volume 18, issue: 10, Page(s): 1810 – 1824, October 2000.

40. Narula-Tam, A., Modiano, E. and Brzezinski, A.; "Physical topology design for survivable routing of logical rings in WDM-based networks," *IEEE GLOBECOM Global Telecommunications Conference,* Volume 5, Page(s): 2552 – 2557, December 2003.

41. Ou, C., Zhu, K., Zang, H., Sahasrabuddhe, L. H. and Mukherjee, B.; "Traffic grooming for survivable WDM networks – shared protection", *Selected Areas in Communications, IEEE Journal on,* Volume 21, Issue 9, Page(s): 1367 – 1383, November 2003.

42. Ozdaglar, A.E., and Bertsekas, D.P.; "Routing and wavelength assignment in optical networks", *IEEE/ACM Transactions on Networking*, Volume 11, Issue 2, Page(s): 259 – 272, April 2003

43. Ramamurthy, S. and Mukherjee, B.; "Survivable WDM mesh networks. Part I – Protection," *IEEE INFOCOM, Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies,* Volume 2, Page(s): 744 – 751, 1999.

44. Ramamurthy, S., Sahasrabuddhe, L. and Mukherjee, B.; "Survivable WDM mesh networks", *Lightwave Technology, Journal of,* Volume 21, Issue 4, Page(s): 870 – 883, April 2003.

45. Ramamurthy, R., and Mukherjee, B.; Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks. IEEE/ACM Transactions on Networking, Volume 10, Issue 3, Page(s): 351 – 367, June 2002.

46. Ramasubramanian, S.; "On failure dependent protection in optical grooming networks". *Proceedings of the International Conference on Dependable Systems and Networks (DSN),* Page(s): 475 – 484, 2004.

47. Ramaswami S. and Mukherjee, B.; "Survivable WDM mesh networks Part II – Restoration", *IEEE International Conference on Communications*, Volume 3, Page(s): 2023 – 2030, 1999.

48. Ramaswami, R. and Sivarajan, K. N.; "Design of logical topologies for wavelength-routed optical networks," *IEEE Journal on Selected Areas in Communications*, Volume 14, Issue 5, Page(s): 840 – 851, June 1996.

49. Ramaswami, R. and Sivarajan, K. N.; "Optical Networks: A Practical Perspective", *Morgan Kaufmann Publishers, ISBN: 1-55860-655-6,* 1998.

50. Rouskas, G. and Dutta, R.; "Design of Logical Topologies for Wavelength Routed Networks – chapter – *Optical WDM Networks: Principles and Practice*, pages 79 – 102, Kluwer, 2000.

51. Rouskas, G.; Dutta, R.; "Design of Logical Topologies for Wavelength Routed Networks," *Optical WDM Networks: Principles and Practice, Kluwer, Norwell, Massachusetts*, Page(s): 79 – 102, 2000.

52. Sahasrabuddhe, L., Ramamurthy, S. and Mukherjee, B.; "Fault Management in IP-Over-WDM Networks: WDM Protection versus IP Restoration", *IEEE Journal on Selected Areas in Communications*, Volume 20, Issue 1, January 2002.

53. Somani, A. K.; "Survivability and traffic grooming in WDM mesh networks", *Cambridge University Press, ISBN 0521853885,* 2006.

54. Stamatelakis, D. and Grover, W. D.; "IP Layer Restoration and Network Planning Based on Virtual Protection Cycles", *IEEE Journal on Selected Areas in Communications.*, Volume18, Issue 10, Page 1938 – 1949, October 2000.

55. Subrata, K. S.; "Heuristic for the Design of Fault Tolerant Logical Topology", *Masters Thesis, School of Computer Science, University of Windsor,* 2005.

56. Thiagarajan, S. and Somani, A. K.; "Traffic grooming for survivable WDM mesh networks". *In OptiComm 2001: Optical Networking and Communications*, Volume 4599, pages 54 – 65, 2001.

57. Thiagarajan, S. and Somani, A. K.; A capacity correlation model for WDM networks with constrained grooming capabilities, *Communications, 2001. ICC 2001. IEEE International Conference on,* Volume 5, Page(s): 1592 – 1596, June 2001.

58. Urquhart, P., Lopez, O. G., Boyen, G. and Bruckmann, A.; "Optical Amplifiers for Telecommunications", *Intelligent Signal Processing, WISP 2007. IEEE International Symposium on,* Page(s): 1 – 6, October 2007

59. Wu, T. H.; "Emerging technologies for fiber network survivability," *IEEE Communication Magazine,* Volume 33, Page(s): 58 – 74, February 1995.

60. Wu, T.; "Fiber Network Service Survivability" *Norwood, MA: Artech House, ISBN: 978-0890064696,* 1992.

61. Xiang, B., Wang, S. and Li, L.; "A traffic grooming algorithm based on shared protection in WDM mesh networks", *Parallel and Distributed Computing,*

Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on, Page(s): 254 – 258, August 2003.

62. Xiang, B., Yu, H., Wang, S. and Li, L.; "A differentiated shared protection algorithm supporting traffic grooming in WDM mesh networks", *Communications, Circuits and Systems, 2004. ICCCAS 2004. 2004 International Conference on,* Volume 1, Page(s): 628 – 632, June 2004.

63. Xin, C. and Qiao, C.; "Performance analysis of multi-hop traffic grooming in mesh WDM optical networks", *Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on,* Page(s): 237 – 242, October 2003.

64. Xin, C., Wang, B., Cao, X. and Li, J.; "Logical topology design for dynamic traffic grooming in WDM optical networks", *Lightwave Technology, Journal of,* Volume 24, Issue 6, Page(s): 2267 – 2275, June 2006.

65. Yao, W. and Ramamurthy B.; "Survivable traffic grooming in wdm mesh networks under SRLG constraints", *Communications, 2005. ICC 2005. 2005 IEEE International Conference on,* Volume 3, Page(s): 1751 – 1755, May 2005.

66. Zang, H., Jue, J. P. and Mukherjee, B.; "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks", *Optical Networks Magazine,* January 2000.

67. Zang, K. and Mukherjee, B.; "A review of traffic grooming in WDM optical networks: Architectures and challenges". *Optical Networks Magazine,* Volume 4, Issue 2, Page(s): 55 – 64, March 2003.

68. Zhang, J. and Mukheriee, B.; "A review of fault management in WDM mesh networks: basic concepts and research challenges," *IEEE Network*, Volume 8, issue 2, Page(s): 41 – 48, April 2004.

69. Zhang, J., Keyao, Z., Sahasrabuddhe, L., Yoo, S. J. B. and Mukherjee, B.; "On the study of routing and wavelength assignment approaches for survivable wavelength routed WDM mesh networks", *Optical Networks Magazine*, November/December 2003.

70. Zheng, J. and Mouftah, H. T.; "Optical WDM Networks: Concepts and Design Principles", *Wiley-IEEE Press,* July 2004.

71. Zhou, D. and Subramaniam, S.; "Survivability in Optical Networks," I*EEE Network*, Volume 14, Issue 6, Page(s): 16 – 23, December 2000.

72. Zhu, H., Zang, H., Zhu, K. and Mukherjee, B.; "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks", *IEEE/ACM Transactions on Networking,* Volume 11, Issue 2, Page(s): 285 – 299, 2003.

73. Zhu, K. and Mukherjee, B.; "Traffic grooming in an optical WDM mesh network", *Selected Areas in Communications, IEEE Journal on,* Volume 20, Issue 1, Page(s): 122 – 133, January 2002.

74. Nippon Telegraph and Telephone Corporation (NTT), Chiyoda Ward, Tokyo, Japan, News Release, September 2006, http://www.ntt.co.jp.

APPENDIX A: ABBREVIATIONS

1. WDM – Wavelength Division Multiplexing
2. DWDM – Dense Wavelength Division Multiplexing
3. RWA - Routing & Wavelength Assignment
4. LP – Linear Programming
5. ILP – Integer Linear Programming
6. MILP – Mixed-Integer Linear Programming
7. ADM – Add Drop Multiplexer
8. OADM – Optical Add Drop Multiplexer
9. OXC - Optical Cross-Connect
10. SOA – Semiconductor Optical Amplifier
11. EDFA – Erbium-Doped Fiber Amplifier
12. LED – Light Emitting Diode
13. WAN - Wide Area Network
14. LAN - Local Area Network

APPENDIX B: TRAFFIC MATRIX

In Our Experiments, we have used 15 different traffic matrices for each of 5 different physical topology of a particular size of network, in which 5 for high traffic, 5 for medium traffic and 5 for low traffic. We list here, for example, three such traffic matrices for 6-nodes networks, one for each category of traffic volume.

1. Traffic Matrix for 6 Node Network (Low Traffic)

Nodes	**0**	**1**	**2**	**3**	**4**	**5**
0	0	OC-3x8	OC-3x3 OC-6x2	OC-3x1 OC-6x3	OC-3x2	OC-3x3
1	OC-3x1	0	OC-3x5	OC-24x1	0	OC-3x1 OC-6x1
2	OC-3x2 OC-6x1	OC-3x4 OC-6x1	0	0	OC-6x1	OC-3x3 OC-6x1
3	OC-3x2	OC-3x5	OC-3x2	0	OC-3x2	OC-24x1
4	OC-6x1	OC-24x1	OC-3x1	OC-3x1	0	OC-3x2
5	OC-12x1	OC-12x1	OC-3x1 OC-6x1 OC-12x1	OC-6x1	OC-6x1	0

2. Traffic Matrix for 6 Node Network (Medium Traffic)

Nodes	0	1	2	3	4	5
0	0	0	OC-3x4	OC-3x1 OC-6x1	OC-3x3 OC-24x1	OC-6x1 OC-24x1
1	0	0	OC-3x1	OC-3x2 OC-24x1	OC-12x1 OC-24x1	OC-3x2
2	OC-3x8 OC-6x4	OC-3x3	0	OC-3x1 OC-6x1 OC-12x2	OC-3x1 OC-24x1	OC-3x1 OC-6x1
3	OC-6x3 OC-12x2	OC-3x1 OC-6x1	OC-3x4 OC-6x1	0	OC-3x1	OC-6x1 OC-12x1
4	OC-3x2 OC-12x1 OC-24x1	OC-6x1 OC-12x1	OC-6x2	OC-3x1 OC-24x1	0	OC-3x5 OC-6x1
5	OC-3x1 OC-24x1	OC-3x1	OC-3x2 OC-6x1 OC-12x2	OC-6x3 OC-12x1	OC-3x1 OC-6x2	0

3. Traffic Matrix for 6 Node Network (High Traffic)

Nodes	0	1	2	3	4	5
0	0	OC-6x10	OC-3x2 OC-6x11	OC-3x4 OC-6x13	OC-3x1 OC-6x3 OC-12x2 OC-24x1	OC-12x6
1	OC-3x5	0	OC-3x1 OC-6x3	OC-3x1 OC-24x3	OC-3x7	OC-3x11 OC-6x1 OC-12x2 OC-24x1
2	OC-3x3	OC-3x2 OC-6x2	0	OC-3x3 OC-6x4 OC-12x3 OC-24x1	OC-3x1 OC-6x1 OC-12x2	OC-3x1 OC-24x3
3	OC-3x5 OC-6x1 OC-24x3	OC-3x1 OC-6x2 OC-12x2 OC-24x1	0	0	OC-24x2	OC-3x3 OC-6x2 OC-24x1
4	OC-3x5 OC-6x2 OC-12x6	OC-3x7 OC-6x2 OC-12x2 OC-24x1	OC-3x5 OC-12x1 OC-24x1	OC-3x2 OC-6x1 OC-12x1 OC-24x2	0	OC-3x4
5	OC-24x2	OC-3x1 OC-12x6	0	OC-3x3	OC-24x3	0

APPENDIX C: COMMODITY TABLE

Commodity table is a data structure, which is used to store list of commodities. We have defined a commodity (Section 3.2) as a triplet consisting of the source s_q, the destination d_q and the quantity q of a data communication request. As an example, we have shown here a single sorted commodity table, which correspond to the traffic matrix for 6-nodes network at low traffic as shown in the Appendix A.

$Q\#$	s_q	d_q	q
1	1	3	24
2	3	5	24
3	4	1	24
4	5	0	12
5	5	1	12
6	5	2	12
7	0	2	6
8	0	2	6
9	0	3	6
10	0	3	6
11	0	3	6
12	1	5	6
13	2	0	6
14	2	1	6
15	2	4	6
16	2	5	6
17	4	0	6
18	5	2	6
19	5	3	6
20	5	4	6
21	0	1	3
22	0	1	3
23	0	1	3
24	0	1	3
25	0	1	3
26	0	1	3
27	0	1	3
28	0	1	3
29	0	2	3
30	0	2	3
31	0	2	3
32	0	3	3
33	0	4	3
34	0	4	3
35	0	5	3

$Q\#$	s_q	d_q	q
36	0	5	3
37	0	5	3
38	1	0	3
39	1	2	3
40	1	2	3
41	1	2	3
42	1	2	3
43	1	2	3
44	1	5	3
45	2	0	3
46	2	0	3
47	2	1	3
48	2	1	3
49	2	1	3
50	2	1	3
51	2	5	3
52	2	5	3
53	2	5	3
54	3	0	3
55	3	0	3
56	3	1	3
57	3	1	3
58	3	1	3
59	3	1	3
60	3	1	3
61	3	2	3
62	3	2	3
63	3	4	3
64	3	4	3
65	4	2	3
66	4	3	3
67	4	5	3
68	4	5	3
69	5	2	3

VITA AUCTORIS

NAME	: Quazi R. Rahman
PLACE OF BIRTH	: Joypurhat, Bangladesh
YEAR OF BIRTH	: 1955
EDUCATION	: Bangladesh University of Engineering and Technology, Dhaka, Bangladesh 1974 – 1978, B. Sc. (Electrical Engg.) The City University of New York New York, New York, USA 1994 – 1997, MS (Computer Science) University of Windsor Windsor, Ontario, Canada 2006 – 2008, M. Sc. (Computer Science)