UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

SURVIVABLE LOGICAL TOPOLOGY MAPPING UNDER MULTIPLE CONSTRAINTS IN IP-OVER-WDM NETWORKS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

TA-CHUN LIN
Norman, Oklahoma
2011

SURVIVABLE LOGICAL TOPOLOGY MAPPING UNDER MULTIPLE CONSTRAINTS IN IP-OVER-WDM NETWORKS

A DISSERTATION APPROVED FOR THE
SCHOOL OF COMPUTER SCIENCE

BY

Dr. Krishnaiyan Thulasiraman, Chair

Dr. Theodore Trafalis

Dr. Sridhar Radhakrishnan

Dr. Changwook Kim

Dr. Qi Cheng

© Copyright by TA-CHUN LIN 2011
All Rights Reserved.

Dedication

I dedicate this dissertation to my family, especially my best friend and beloved wife Zhili Zhou who has put up with me all these years and shared the uncertainties, challenges and sacrifices for completing this dissertation. My appreciation goes to my loving parents, Pennan Lin and Pihua Weng, and terrific in-laws who have been proud and supportive of my work. I must also thank my grandaunt, my sister Yiching Lin, and my brother-in-law Yuyi Hsiao who believed in diligence and have never left my side. Finally, I would like to dedicate this work to the memory of my dear grandparents who live forever in my heart.

Acknowledgements

Firstly and most importantly, I would like to express my deepest gratitude and sincere appreciation to my advisor, Dr. Krishnaiyan Thulasiraman. Without his guidance, inspiration, and support throughout the journey, this dissertation would not be possible. His boundless enthusiasm, persistent commitment to high quality research, and infinite patience were essential for the completion of this dissertation. I feel exceedingly fortunate to have had Dr. Thulasiraman as my mentor and friend. I hope that, one day, I will be able to become an advisor who is just as helpful and dedicated as he is.

I greatly appreciate the advice and support I have received from my dissertation committee members: Dr. Theodore Trafalis, Dr. Sridhar Radhakrishnan, Dr. Changwook Kim, and Dr. Qi Cheng. Indeed, I am very honored to have had such a highly prestigious committee with exceptional expertise in Theoretical Computer Science, Networking, and Operations Research.

I further like to thank Dr. S. Lakshmivarahan, Dr. Carl Sondergeld, Dr. Chandra Rai, Dr. Henry Neeman, Dr. Michele Eodice, Dr. Duane Stock, Dr. Bryan Stanhouse, Moira Ozias, and Tara Nielsen. It was my great pleasure to work with you as a teaching/research assistant and I appreciate your support during my years at the University of Oklahoma.

My special gratitude goes to my friends: Yuh-Rong Chen, Muhammad Javed, Aravind Mohanoor, Shirley Hsieh, Pedro Diaz-Gomez, Shohrab Hossain, Amlan Chatterjee, Chih-Yu Lai, Tim Huang, Toni Chen for stimulating discussions and readiness to help.

My special thanks are extended to the wonderful staff members in Computer

Science at the University of Oklahoma, especially Barbara Bledsoe, Chyrl Yerdon, and Jim Summers for the timely help, encouragement, and advice from the first day I entered the doctoral program.

Last but not least, I sincerely appreciate: my former major advisor Dr. Feng-Jian Wang at National Chiao Tung University (NCTU); my former supervisors Dr. Shiuhpyng Shieh at NCTU and Dr. Chi-Chuan Cheng at Chinese Culture University; each and everyone in my family for their boundless love and patience; and numerous others not listed here for their support and friendship.

Table of Contents

List of Tables

List of Figures

Abstract

The survivable logical topology mapping problem in an IP-over-WDM network deals with the cascading effect of link failures from the bottom (physical) layer to the upper (logical) layer. Multiple logical links may get disconnected due to a single physical link failure, which may cause the disconnection of the logical network. Here we study survivability issues in IP-over-WDM networks with respect to various criteria.

We first give an overview of the two major lines of pioneering works for the survivable design problem. Though theoretically elegant, the first approach which uses Integer Linear Programming (ILP) formulations suffers from the drawback of scalability. The second approach, the structural approach, utilizes the concept of duality between circuits and cutsets in a graph and is based on an algorithmic framework called Survivable Mapping Algorithm by Ring Trimming (SMART). Several SMART-based algorithms have been proposed in the literature.

In order to generate the survivable routing, the SMART-based algorithms require the existence of disjoint lightpaths for certain groups of logical links in the physical topology, which might not always exist. Therefore, we propose in Chapter 4 an approach to augment the logical topology with new logical links to guarantee survivability. We first identify a logical topology that admits a survivable mapping against one physical link failure. We then generalize these results to achieve augmentation of a given logical topology to survive multiple physical link failures.

We propose in Chapter 5 a generalized version of SMART-based algorithms and introduce the concept of robustness of an algorithm which captures the ability of the algorithm to provide survivability against multiple physical link failures. We demonstrate that even when a SMART-based algorithm cannot be guaranteed to

provide survivability against multiple physical link failures, its robustness could be very high.

Most previous works on the survivable logical topology design problem in IP-over-WDM networks did not consider physical capacities and logical demands. In Chapter 6, we study this problem taking into account logical link demands and physical link capacities. We define weak survivability and strong survivability in capacitated IP-over-WDM networks. Two-stage Mixed-Integer Linear Programming (MILP) formulations and heuristics to solve the survivable design problems are proposed. Based on the 2-stage MILP framework, we also propose several extensions to the weakly survivable design problem, considering several performance criteria. Noting that for some logical networks a survivable mapping may not exist, which prohibits us from applying the 2-stage MILP approach, our first extension is to augment the logical network using an MILP formulation to guarantee the existence of a survivable routing. We then propose approaches to balance the logical demands satisfying absolute or ratio-weighted fairness. Finally we show how to formulate the survivable logical topology design problem as an MILP for the multiple failure case. We conclude with an outline of two promising new directions of research.

Chapter 1

Introduction

1.1 Overview

When ARPANET was initiated in 1968, people would not have envisioned that the Internet would grow to the scale we have today. The increasing communication demands due to bandwidth-intensive applications such as video streaming have pushed the development toward the design of high bandwidth, less error-prone, and more cost- and power-efficient networks.

Optical fiber is broadly considered to be the transmission medium for the next generation communication network because of its high bandwidth, low bit-error rate, and cost. The first generation optical fiber networks, for example, SONET (synchronous optical network) and SDH (synchronous digital network), were interconnected with optical fibers, where the optics served the purpose of transmission and the routing mechanism over such networks relied on electrical routers, switches, etc., which have no capability of processing the optical information. Hence each node needs to process the information which is either targeted to the node or just transmitted through the node to the destination. So opto-electronic-optic (O-E-O) conversion was required on each hop, which introduced extra signalling and processing overhead generated between layers. The overhead affected the processing speed and scalability of the optical fiber networks.

The concept of all-optical networks (AON) combined with the development of optical add-drop multiplexers (OADM) and optical cross-connects (OXC) eliminates the requirement of O-E-O conversions on intermediate nodes. This second generation optical network has the routing and switching capability and hence it can deal with point-to-point transmission as well as multi-hop communication. For multi-hop communication, electrical signal is first converted to optical signal and such information is passed through intermediate nodes without the O-E-O conversion. Once it reaches the destination, the optical signal will then be converted back to electrical signal for further processing in the higher layer.

In early days each optical fiber carried only a single wavelength. To better utilize the capacity of the wavelength researchers proposed Optical Time-Division Multiplexing (OTDM) technique where electrical data was multiplexed in extremely narrow optical timeslots. However, OTDM faces issues such as optical signal degradation in the optical fiber and devices along the transmission path (called impairments). The introduction of the Wavelength-Division Multiplexing (WDM) technique, a frequency-division multiplexing (FDM) technique, brings the ultra-high capacity to optical fibers where the bandwidth in an optical fiber is divided into segments, each carrying individual signals. Thus with the WDM technique there are multiple *logical/virtual fibers*, each carrying individual data streams that do not interfere with one another, for each physical optical fiber. With the advances in optics and photonics and WDM techniques, multiple independent data channels can be multiplexed on a single fiber over different wavelengths (equivalently, colors or frequencies).

Recently it was demonstrated that an optical data transmission rate of up to 109 terabits per second (Tb/s) within an optical fiber [1] can be achieved for a distance of 16.8km, which exceeds last year's record of 69.1 Tb/s over a single 240 km-long optical fiber [2] and more than quadruple the maximum transmission rate in year 2008 [3].

In the early 1980s the International Standards Organization (ISO) proposed a layered network architecture. Optical networks, which support multiple protocols such as Internet Protocol (IP), Asynchronous Transfer Mode (ATM), and SONET, can actually be treated as an additional *optical layer* providing services to supported upper layer protocols. Early implementation of WDM networks applied multi-layer protocol stack between IP and WDM layers, for example, IP over ATM over SONET over WDM network, where the protocol overhead between layers may occupy 22% of the bandwidth [4]. In addition, each layer has different transmission rate and each of them has its protection/restoration and routing mechanism which further reduces the efficiency and transmission rate provided by the optical medium. Since TCP/IP is the most dominating protocol stack nowadays, implementing IP protocol directly over an optical network with WDM channel support, which reduces the routing, signalling and the delay incurred between layers, has caught the attention of researchers. This kind of network is usually called the IP-over-WDM network.

A commonly proposed approach to implement an IP network over the WDM network is to embed an IP topology, referred to as the logical topology, into a WDM topology, called the physical topology. The mapping involves finding a routing for the corresponding end nodes (source and destination) of an IP (logical) link in the physical topology. Such kind of a path is called a *lightpath*, which is an all-optical path in the physical topology established by allocating a wavelength between the corresponding terminal nodes of an IP link. A lightpath, once established, does not require processing or buffering at intermediate nodes and may bypass intermediate O-E-O conversions (which are required in the first-generation optical networks).

In an IP-over-WDM network, a single fiber generally carries several lightpaths simultaneously and usually several fibers are bundled together to form a cable (a physical link). Therefore, a cable cut in the network can disrupt all the lightpaths passing through the cable and degrade the network performance significantly, if the failure persists. Unfortunately, cable cuts and equipment failures have become a common occurrence due to human or natural events, drawing considerable attention to designing networks that can provide an acceptable level of service in the presence of such failures. Such networks are generally called survivable networks.

Protection and restoration are the two widely discussed mechanisms for providing survivability in IP-over-WDM networks. Protection in IP-over-WDM networks is generally provided at the physical layer at the design stage. First, primary or working lightpaths are established for the logical links and then backup or protection lightpaths are calculated that do not use physical links (or nodes) already assigned to their respective primary lightpaths (i.e., their mappings are disjoint). In case of a failure, the network traffic carried by a primary lightpath is always switched to its corresponding backup lightpath. Since protection requires explicit reservation of resources, it is generally very fast but inefficient in terms of resource utilization.

Restoration is generally provided at the logical layer by provisioning the network with some additional capacity, which can be utilized by the IP routers to find backup paths after a failure. It is possible to find backup paths only if the logical links are mapped onto the physical topology in such a way that the logical topology remains connected after the failure. This can be achieved by requiring the logical and physical topologies to be at least 2-edge connected and finding link/node-disjoint mappings for some or all the logical links. A mapping of the logical links so that the logical network remains connected after the failure of physical link/links is called a link-survivable mapping and a mapping that stays connected after the failure of physical node/nodes is called a node-survivable mapping.

The problem of finding link/node-survivable mappings is known to be NP-complete [5]. Therefore, efficient algorithms to solve the problem are unlikely in full generality.

1.2 WDM Optical Network and Survivability Mechanisms

In this section, we provide a brief introduction to WDM networks and discuss some commonly used WDM architectures. We also introduce the concept of survivable networks, networks that provide an acceptable level of service in the presence of failures, and discuss various mechanisms that can be employed to make a given network survivable.

1.2.1 WDM Optical Network

Optical fibers possess several properties which made them the ideal replacement of copper cables in the traditional telephone networks. Optical fibers are not only low-cost, lightweight, and difficult to wiretap, but they also offer very high bit rates (up to 160 Gbps), better signal quality (optical fibers have an approximate Bit Error Rate (BER) of 10^{-12} to 10^{-14} compared to 10^{-3} to 10^{-4} for copper wires), and are immune to electro-magnetic and radio-frequency interference (EMI/RFI) [6].

Initially, optical fibers were mainly used as transmission links by phone companies to upgrade their trunk lines from copper wires. Trunk lines are always digital and employ time division multiplexing (TDM) to support several simultaneous voice connections. These pure point-to-point systems or networks are the simplest form of optical networks and are usually set up using an optical transmitter, a fiber and an optical receiver. An optical transmitter is essentially a light source that converts data into a sequence of on/off light pulses of a particular wavelength (λ), which travel through the optical fiber and arrive at the receiver. The receiver then converts the light pulses back into data. Figure 1.1 shows one such network.

With the advent of the WDM technology, the phone companies switched to coarse or dense wavelength division multiplexing (CWDM and DWDM, respectively) to improve transmission speeds and capacity.

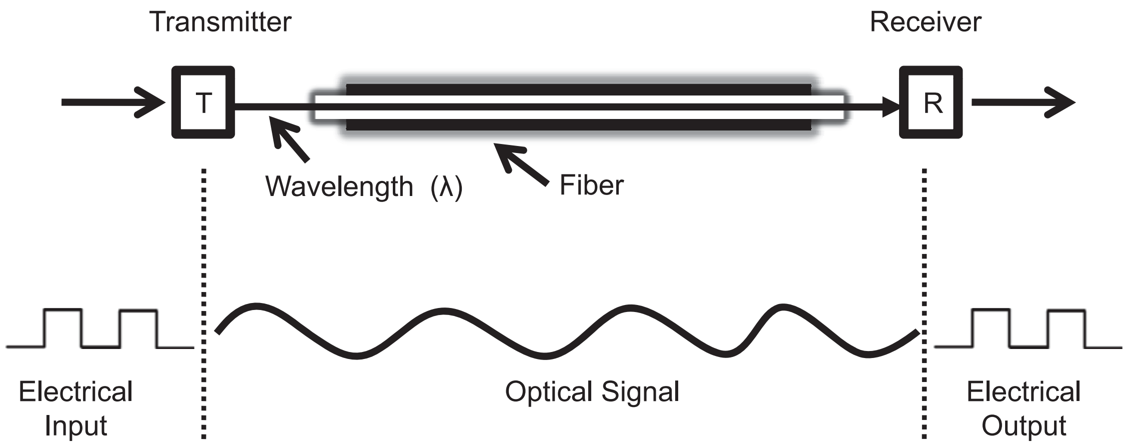


Figure 1.1: A basic optical network

WDM technology allows a single fiber to simultaneously carry multiple optical signals (i.e., channels), each modulating at a unique wavelength. A wavelength can be thought of as a different color of light in the infrared spectrum that can carry data. Since the number of wavelengths that a fiber can carry is generally limited by the end equipment (e.g., transmitters, receivers, multiplexer/de-multiplexer, etc.) not by the fiber itself, optical fiber networks employing WDM technology offer unprecedented scalability. Such networks are usually called WDM optical networks or simply WDM networks.

Early commercial WDM networks were point-to-point networks that appeared in 1995. They were based on 2.5 Gbps per wavelength with 8 or 16 available wavelengths and did not require regeneration of signal up to a distance of 750 miles [7]. These are considered the first generation WDM networks and required manual connection set up. Figure 1.2 shows a basic WDM optical network and Fig. 1.3 shows two point-to-point WDM networks connecting three facilities. Such networks cannot perform network operations in optical domain. Therefore, an O-E-O conversion must be performed if switching signal is required. This phenomenon, generally called electrical bottleneck, limits the throughput of the network to rates compatible with the electronic circuitry of the switching equipment.

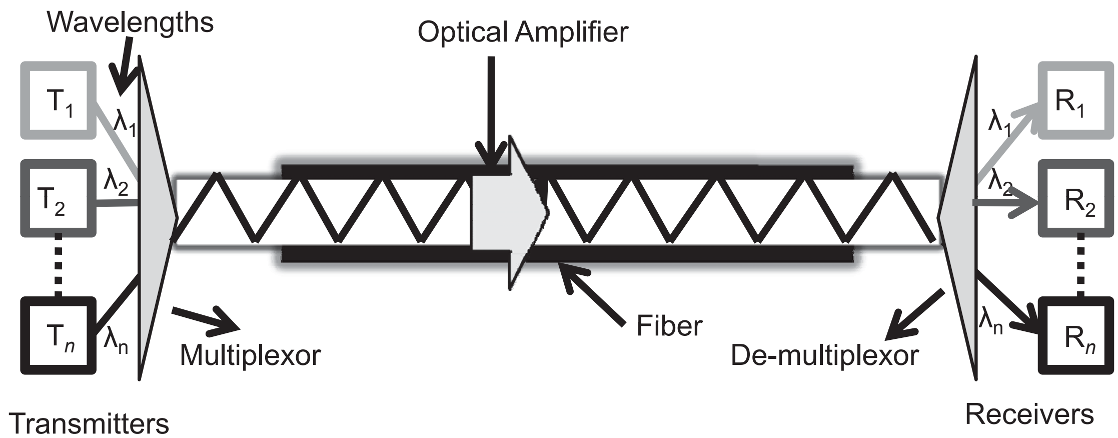


Figure 1.2: A basic WDM optical network

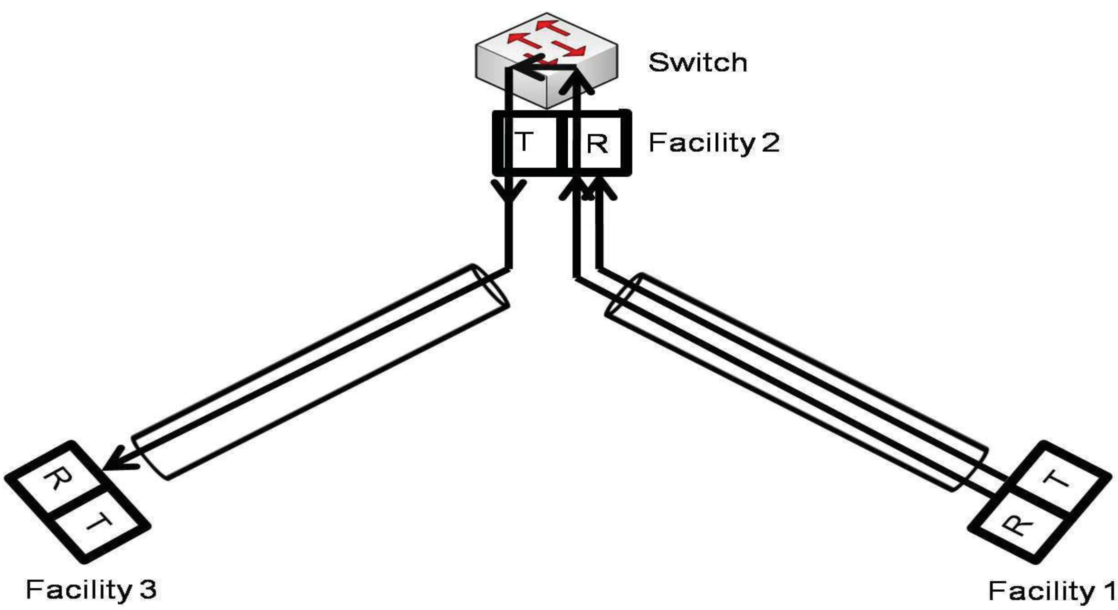


Figure 1.3: Three facilities connected by two point-to-point WDM networks

The popularity of the packet switched World Wide Web (WWW) or Internet in the 1980s and 1990s created a tremendous appetite for more capacity and speed. Given the fact that it is extremely costly to install new fibers to increase transmission capacity, the telecommunication research community focused on increasing the number of wavelengths a fiber can carry and the bit rate. In 1998, the second generation of WDM networks replaced the first generation networks that were characterized by 10 Gbps channels, 40 channels per fiber and semi-automatic connection set up [7]. Such networks used OADMs to provide limited networking functionality. However, OADMs can be used only in point-to-point or ring networks to add or drop signals and have the ability to remove the electrical bottleneck to some extent as shown in Fig. 1.4.

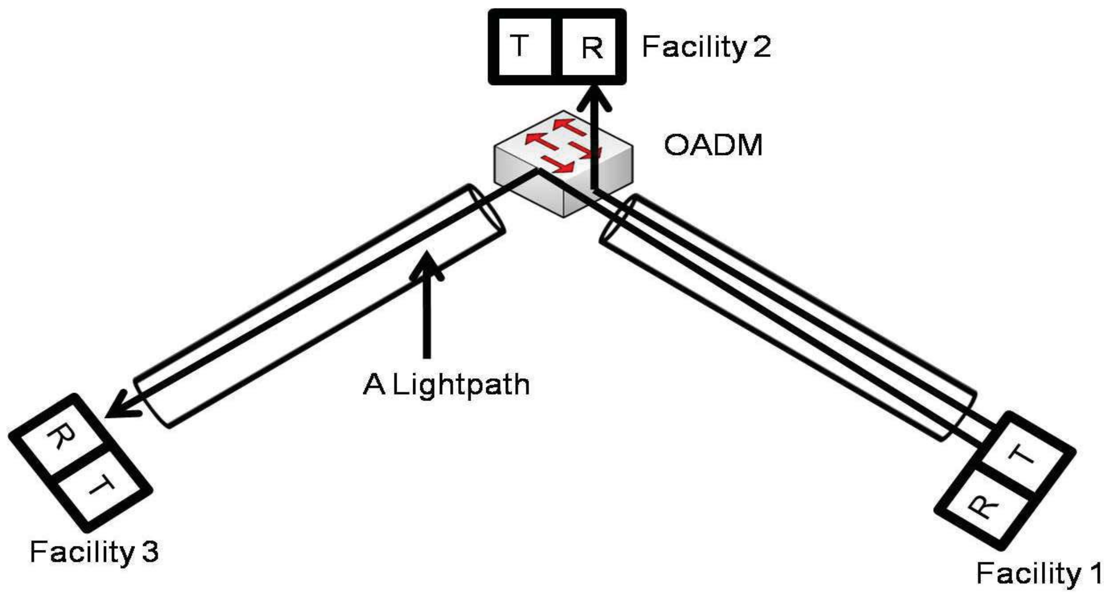


Figure 1.4: Three facilities connected by two point-to-point WDM networks

Introduction of OADMs in WDM networks led to the introduction of lightpath communications [8]. A lightpath is an all-optical path in the WDM network, which is established by the allocation of a particular wavelength between a pair of facilities that may or may not be adjacent to each other. Once established, a lightpath may traverse through multiple fibers without requiring buffering, processing, and quite possibly no O-E-O conversion at the intermediate facilities [8]. However, in some cases it may not be possible to avoid the O-E-O conversion. This may occur when a wavelength is not available to connect a pair of facilities.

The ever-increasing demand for more bandwidth has kept the focus of the research community on developing new types of fibers and enabling networking equipment. Prevailing experimental technologies allow 160 signals per fiber, each modulating at 160 Gbps and providing a total bandwidth of 25.6 Tb/s over three 80 km long single fiber strands without signal regeneration or amplification [3]. Since a fiber optic cable generally contains several hundred fiber strands, an aggregate throughput of several thousand terabits per second can be achieved. Additionally, the development of OXC, optical amplifiers (OA), tunable transmitters/receivers, wavelength converters (WC), etc., now allow more flexible network configurations such as ring and mesh networks. Figure 1.5 and 1.6 show a ring and mesh WDM networks, respectively.

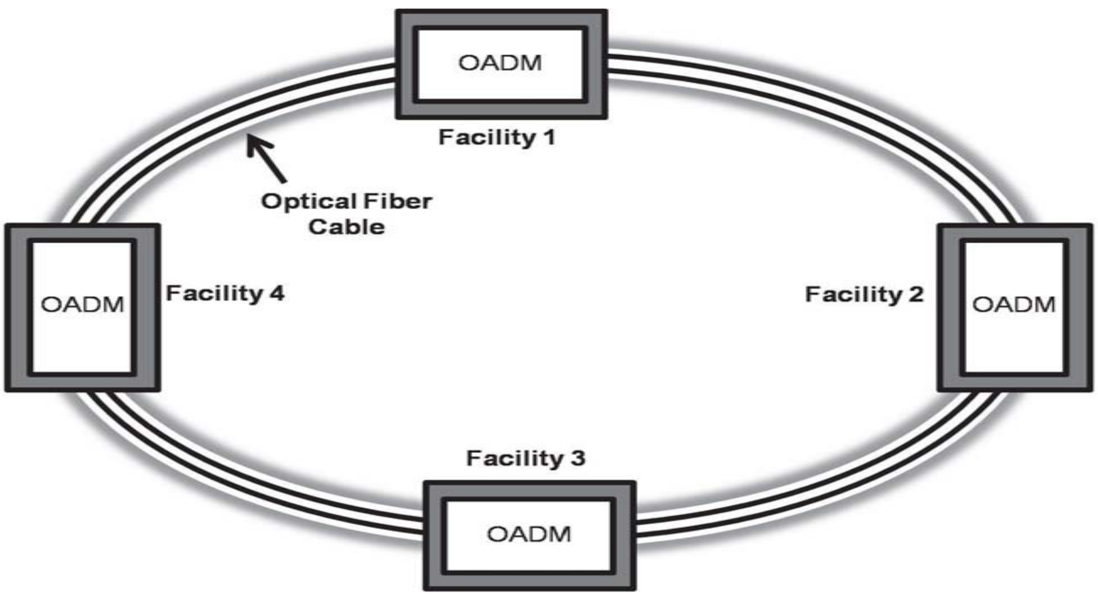


Figure 1.5: A point-to-point WDM network with OADM

An optical cross-connect can be used to add and drop signals as well as to switch traffic from one fiber to another without any O-E-O conversion and a wavelength converter converts a signal at one wavelength to another without O-E-O conversion. OXCs and WCs, when used together in a WDM network, remove the need for O-E-O conversions. Such WDM networks are also called AONs.

Modern WDM networks utilizing all optical networking technologies to provide

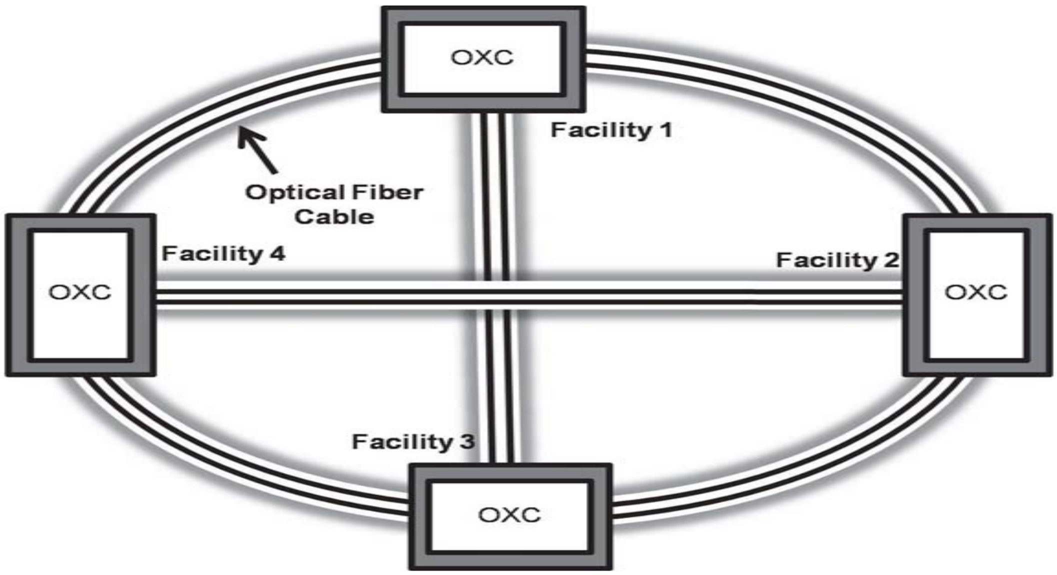


Figure 1.6: A ring WDM network

tremendous bandwidth have posed several challenges to network designers and operators. One such challenge is to determine the best way to effectively utilize the tremendous bandwidth available using existing networking protocols such as IP, ATM, and SONET/SDH. The widespread use of these protocols makes it very difficult to modify them or add new functionalities. Therefore, the most commonly proposed approach to effectively exploit the high bandwidth WDM networks is a layered approach, in which the WDM layer is transparent or invisible to protocols such as IP, ATM, and SONET/SDH.

In the layered approach optical fiber cables, OXCs, OADMs, OAs, etc., form the physical network or the WDM layer and IP routers, ATM switches, SONET/SDH rings, etc., represent the logical network or layer. In the physical network all the network operations, i.e., switching, routing, amplification, etc., are performed in the optical domain. However, a logical layer accomplishes the network operations using an equipment that employs electronic circuitry. Figure 1.7 shows only a few possible combinations of logical and physical layers that can be used to exploit WDM networks [9] and Fig. 1.8 shows an example of the protocol stack shown in Fig. 1.7a.

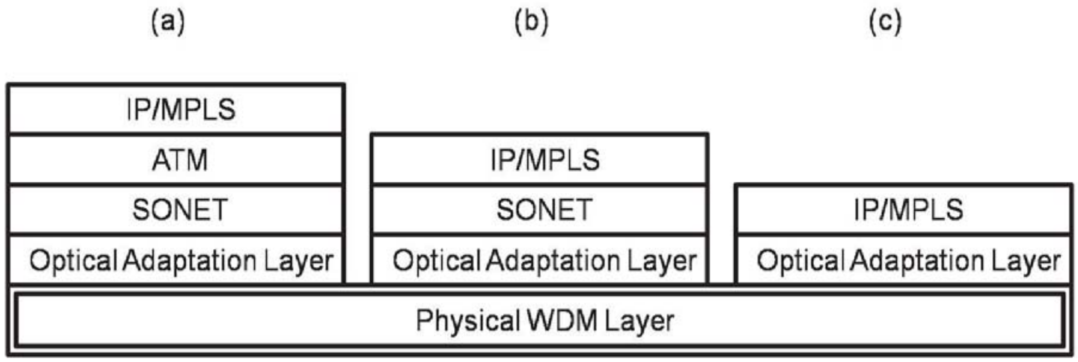


Figure 1.7: Protocol stacks for WDM networks

Another challenge that the network operators and designers must address is the massive amount of data loss that may occur after the failure of network component(s). Usually, several hundred optical fiber strands are bundled together to form a cable. Such cables are then buried along with other utility services like cable, telephone, water, etc., to connect remote facilities. These utility services require continuous maintenance and upgrades, thereby exposing the fiber optic cables to damage and failure. Other networking components like OXCs and OAs may also fail or malfunction but this situation can be remedied by providing redundant equipment. However, it is much more difficult to address fiber failures, which consequently, has drawn more attention from researchers.

The most frequent cause of fiber failures in WDM networks is a fiber cut due to human (construction/repair work, vandalism, etc.) and natural (earthquakes, lightning, fire, etc.) events. Furthermore, the time required to precisely determine the location of the cut and digging up of the cable to perform the repairs is usually significant. Therefore, given the fact that a single fiber may carry enormous amount of data which is lost in the event of its failure and the significant time required to repair the failure, even a single fiber failure can affect the performance of the entire network.

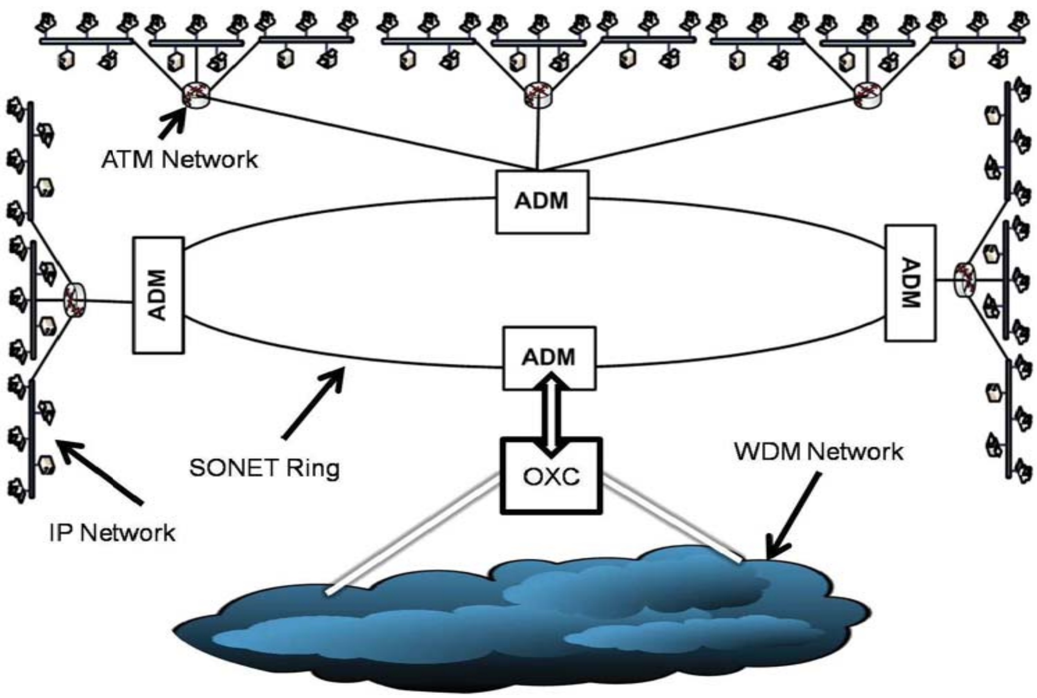


Figure 1.8: An IP-over-ATM-over-SONET-over-WDM network.

The resulting significant degradation in performance can be mitigated by employing mechanisms that allow WDM networks to provide an acceptable level of service in the presence of a failure (or failures). WDM networks with built-in mechanisms that allow them to continue to deliver an acceptable level of service in the presence of a fiber failure are generally called *link-survivable WDM networks*. Similarly, node-survivable WDM networks are able to provide an acceptable level of service after the failure of networking equipment such as OXCs, OAs, etc.

1.2.2 Link-Survivable WDM Network

Link-survivability mechanisms are broadly classified into two categories, namely protection and restoration [6]. Protection is a pre-planned proactive mechanism in which network resources such as fibers, transmitters/receivers, wavelengths, and routers, are explicitly reserved for various failure scenarios. When a fiber fails, the network traffic carried by the affected fiber is simply switched to the resources reserved for this failure scenario. Protection is very fast (on the order of milliseconds) but inefficient in terms of capacity utilization as the reserved resources are idle in the absence of a failure [6].

On the contrary, restoration is a reactive mechanism in which no network resources are set aside for various failure scenarios but the network is provisioned with some extra capacity which can be used to carry the failed fiber network traffic [6]. In restoration, the selection of network resources to be used in the event of a failure is made after the failure has taken place. Restoration is generally considered slow (usually on the order of a few seconds) but more resource-efficient [6]. It is important to note that restoration and protection mechanisms can be implemented only if a network has some degree of redundancy built into it. As mentioned earlier, WDM networks can be subdivided into physical and logical layers, which allows the flexibility to implement survivability mechanisms at physical layer or logical layer or both. Protection is generally associated with the WDM layer. However, it is expected that advances in the development of WDM routers will allow the flexibility to implement restoration at the WDM layer. Logical layer can employ both protection and restoration, but restoration is the preferred mechanism for logical layer.

1.2.3 WDM Layer Survivability Mechanisms

Protection mechanisms are more commonly employed at the WDM layer and greatly depend on the configuration of the network under consideration (e.g., point-to-point, rings, and mesh.) This section provides a brief overview of the WDM layer survivability mechanisms [6]: point-to-point networks, ring networks, and mesh networks.

First, we review the point-to-point network. In point-to-point networks, *automatic protection switching* (APS) is the most widely accepted protection mechanism. APS requires spare fibers which must be buried along a route not used by the main fibers.

The main fibers are generally referred to as primary or working links and the spare fibers are called protection or backup links. There are three main types of APS systems, namely $1:1$, $1+1$ and $1:N$ APS [6].

In $1:1$ APS, the normal network traffic is carried by a primary link and after the failure the traffic is switched to the protection link. In some cases, the protection link may carry low priority traffic which is preempted when the working link fails. Figure 1.9 shows a $1:1$ APS point-to-point network.

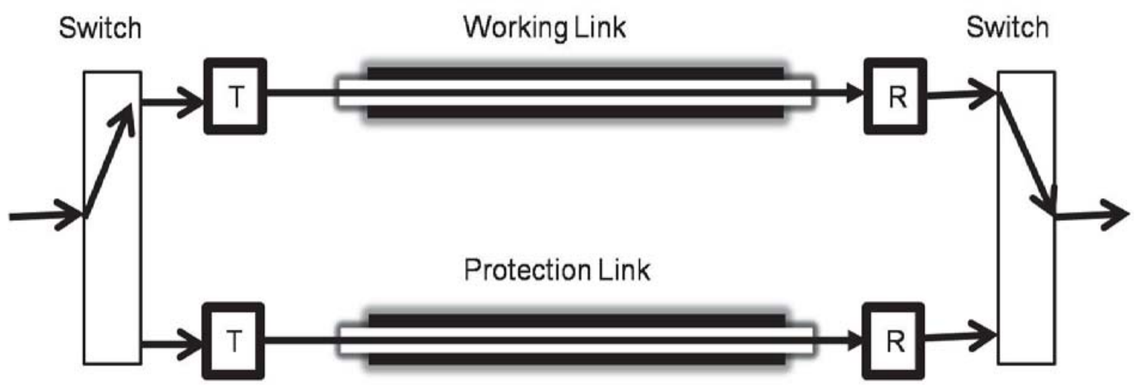


Figure 1.9: 1:1 APS point-to-point network

In $1+1$ APS, the normal network traffic is carried by a primary link and an exact copy of this traffic is also carried on the protection link. The receiver chooses the signal of better quality. Figure 1.10 shows a $1+1$ APS point-to-point network.

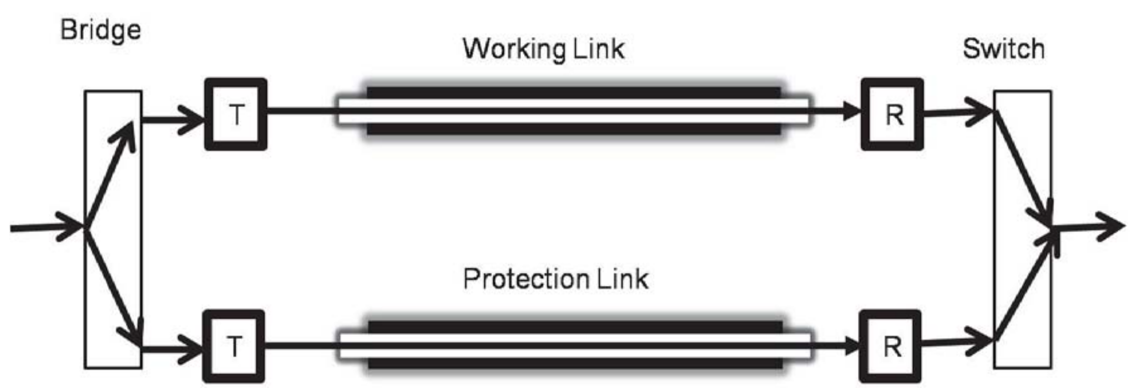


Figure 1.10: 1+1 APS point-to-point network

In $1:N$ APS, one protection link is shared by many working links that are not expected to fail at the same time. When a working link fails, the traffic carried by this link is switched to the protection link. The traffic is switched back to the working link, when it recovers from the failure. Figure 1.11 shows a $1:N$ APS point-to-point network.

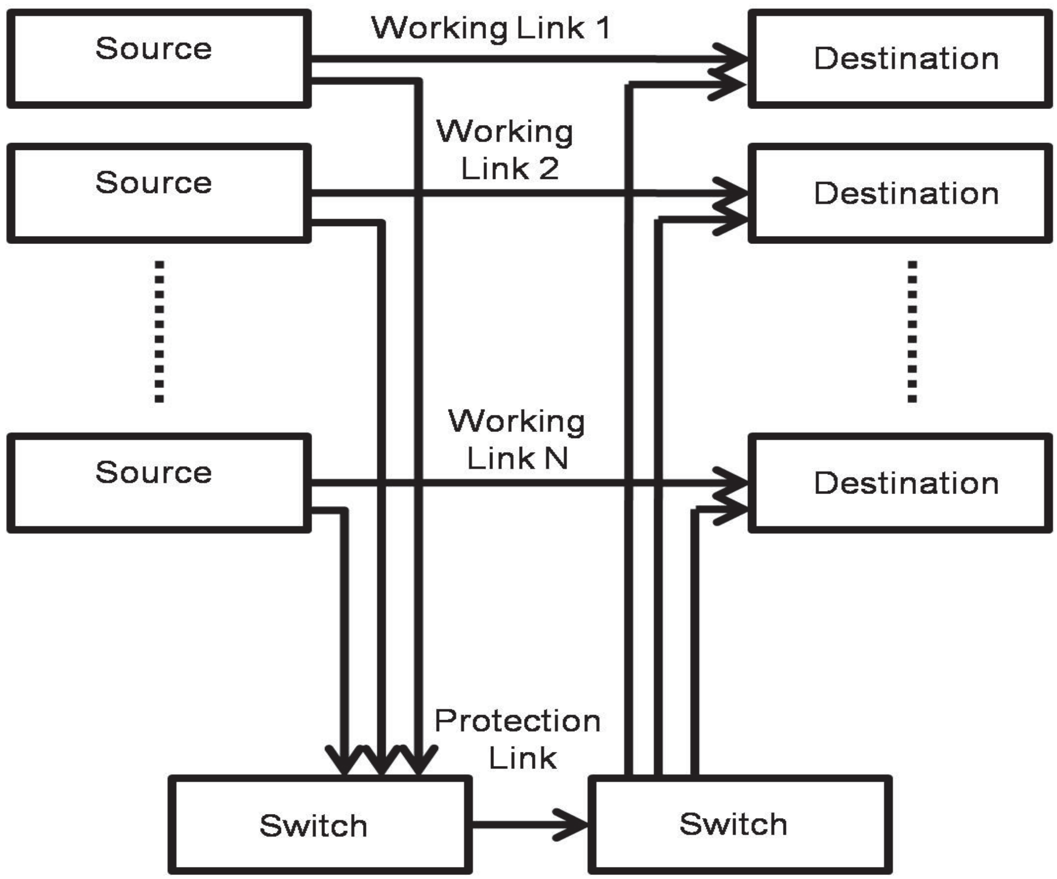


Figure 1.11: 1:N APS point-to-point network

We shall review now the ring network. The concept of APS has also been used in ring networks to design *Self Healing Rings* (SHRs), which protect the networks designed in the form of a ring. Figure 1.12 shows a SHR with two fibers. The working traffic flows in one direction (clockwise or anti-clockwise) on a fiber and in case of a failure the working traffic carried by the failed link flows in the opposite direction on the protection fiber (similar to 1 : 1 APS). In some cases, the signal is transmitted on both working and protection fibers and the nodes decide which signal to choose (similar to $1 + 1$ APS).

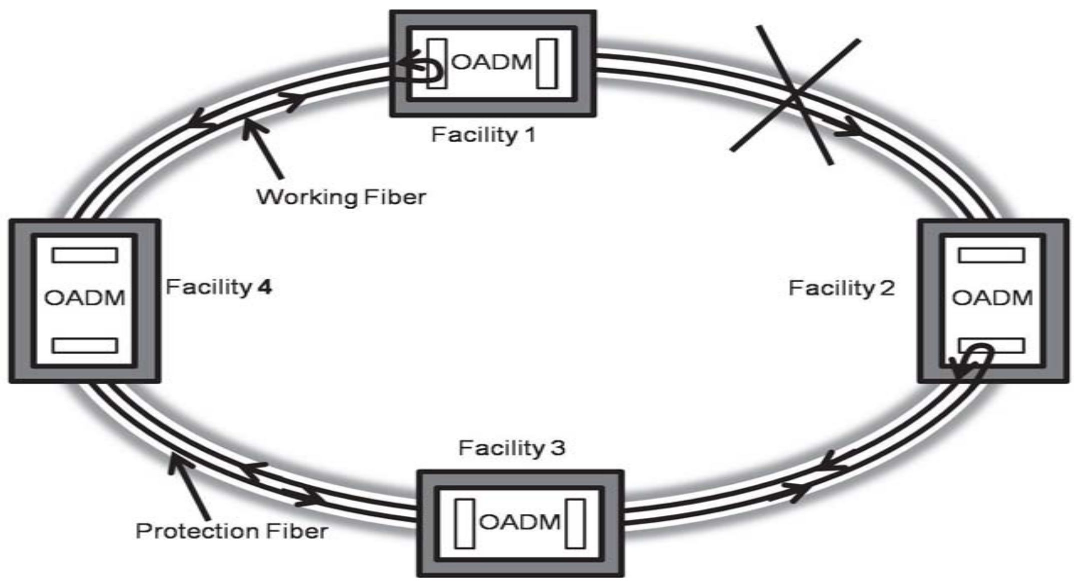


Figure 1.12: Protection in a ring network using a SHR

These methods require that the spare capacity reserved must be equal to the working capacity of the network, which makes the network very inefficient but the time to recover from a failure is negligible, usually on the order of 50 ms [6].

Thirdly, we review the mesh network. Mesh networks have higher link diversity that could lead to lower bandwidth redundancy but the problem of designing fast protection mechanisms becomes more difficult. Various preplanned protection techniques have been proposed for mesh networks. One such technique is Pure Ring Covers (PRCs) that finds multiple logical rings to cover all the links which then work as a collection of SHRs. Figure 1.13 provides an example of PRC. However, PRCs require at least 100% redundancy but in real networks it is sometimes more than 200% [6].

To reduce redundancy, the concept of *Pre-configured Protection Cycles* (p-Cycles) was introduced in [10] to protect mesh networks against a single link failure. p-Cycles require significantly less spare capacity than the other protection mechanisms for mesh networks. p-Cycles typically require a redundancy of $50 - 70\%$ in well-connected physical networks. p-Cycles are based on the ability of a ring to protect not only the links that form the ring but also any possible straddling links. A straddling link is a link which is not part of the ring but its end points lie on the ring. Figure 1.14a shows a p-Cycle that provides a single protection path with ten cycle links and two paths for the eight straddling links. Figure 1.14b shows the behavior of the p-Cycle when a link on the cycle fails. In this case, the p-Cycle behaves like a SHR. Figures 1.14c and 1.14d show the behavior of the p-Cycle when a straddling link fails. In these cases two paths are available to protect the failed links.

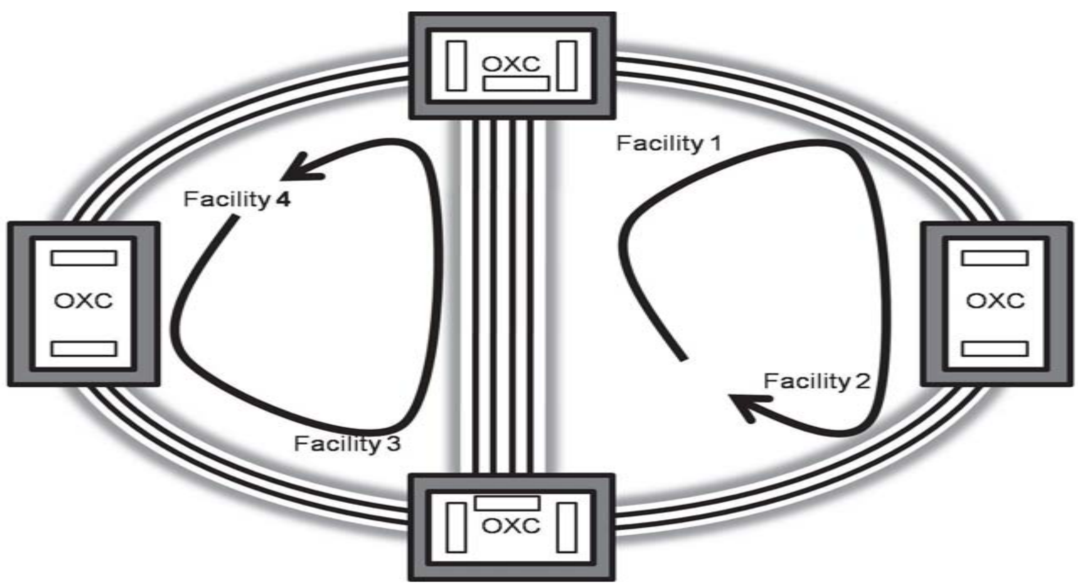
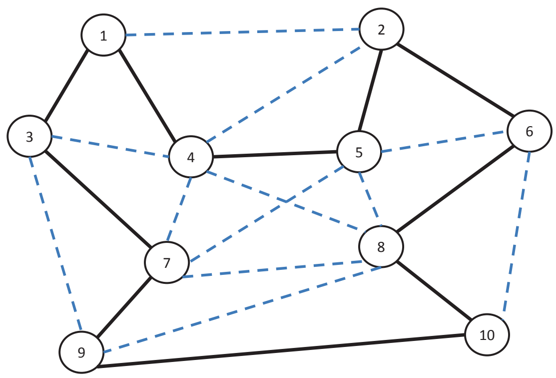
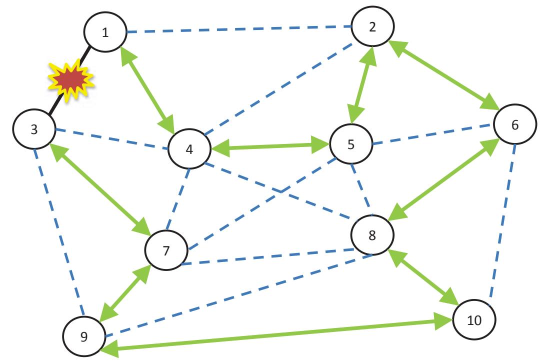


Figure 1.13: A mesh network protected by two rings (PRCs)

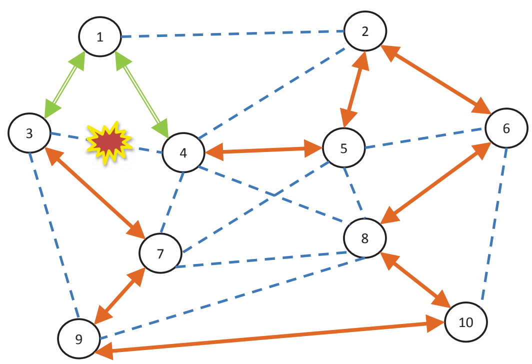
By using a set of carefully designed p-Cycles, it is possible to protect all the links in a mesh network. To minimize the spare to working resources ratio, most of the methods proposed in the literature to design p-Cycles are based on ILPs. In fact the problem of finding a set of p-Cycles that minimizes the spare to working resource



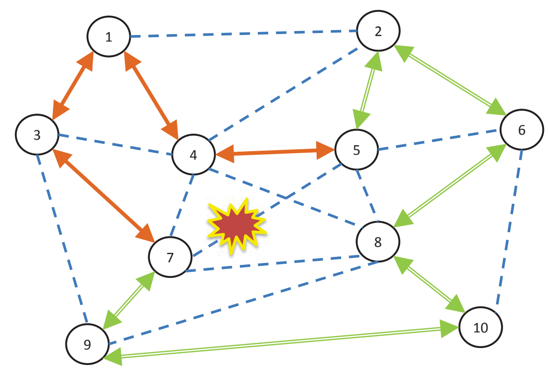
(a) A p-Cycle (bold line)



(b) p-Cycle contributes one restoration path when a link on the p-Cycle (1, 3) fails



(c) p-Cycle contributes two restoration paths when a straddling link (3, 4) fails



(d) p-Cycle contributes two restoration paths when a straddling link (5, 7) fails

Figure 1.14: p-Cycle

ratio is NP-complete [11].

1.2.4 Logical Layer Survivability Mechanisms

As shown in Figure 1.7, the logical layer may consist of several different protocols, which have well developed survivability mechanisms. Some of these mechanisms, i.e., *Transmission Control Protocol/ Internet Protocol* (TCP/IP), ATM, and SONET are discussed below.

TCP/IP networks have a built-in mechanism to reroute traffic around a failed network component through use of various routing protocols such as *Routing Information Protocol* (RIP), *Open Shortest Path First* (OSPF), etc. In TCP/IP networks, after detecting a failure, the IP routers compute the alternate routes for the affected traffic based on the network topology after the failure.

ATM is a connection-oriented protocol. Therefore, a connection must be established before data transmission could begin. Such a connection is usually called the *primary* or *working virtual path* (VP). Restoration is provided in an ATM network by calculating a backup VP for the failed working VP after a failure [12]. The backup VP is selected in such a way that it avoids the failed network component. It is also possible to provide protection in ATM networks by pre-computing the backup VPs for the working VPs such that a failure does not affect both the working and the backup VPs [13].

A SONET is an optical network that is set up using digital cross-connect switches (DCS). The DCSs are responsible for switching, failure detection and restoration in SONET. In case of a failure, the DCSs dynamically establish alternate paths for the traffic on the failed link by utilizing the available spare capacity.

1.3 IP-over-WDM Networks

In this section, we review the IP-over-WDM networks and survivability issues in these networks.

The IP-over-WDM network is a two-layered network where an IP network (*logical network*) is embedded onto a WDM network (*physical network*). IP routers and OXCs correspond to the *logical nodes* and *physical nodes*. Links connecting the nodes in a logical network are called the *logical links*, and the *physical links* are realized via optical fibers. The logical nodes are commonly assumed to have corresponding nodes in the physical network. On the other hand, not all physical nodes may exist in the logical network. A router-to-router link is implemented through a wavelength on a path between two end nodes in a WDM network bypassing O-E-O conversions on intermediate nodes in the path. This path is called a *lightpath*. The *mapping* of a logical topology into the physical topology involves finding a lightpath for each logical link in the physical network. Each optical fiber may carry multiple lightpaths, hence a failure on an optical fiber may have a cascading effect causing failures on multiple logical links, resulting in a huge amount of data traffic loss. This has given rise to an extensive interest in the study of survivability issues in the IP-over-WDM network.

Figure 1.15 illustrates an IP-over-WDM network. In the next section, we review the survivability issues in the IP-over-WDM networks.

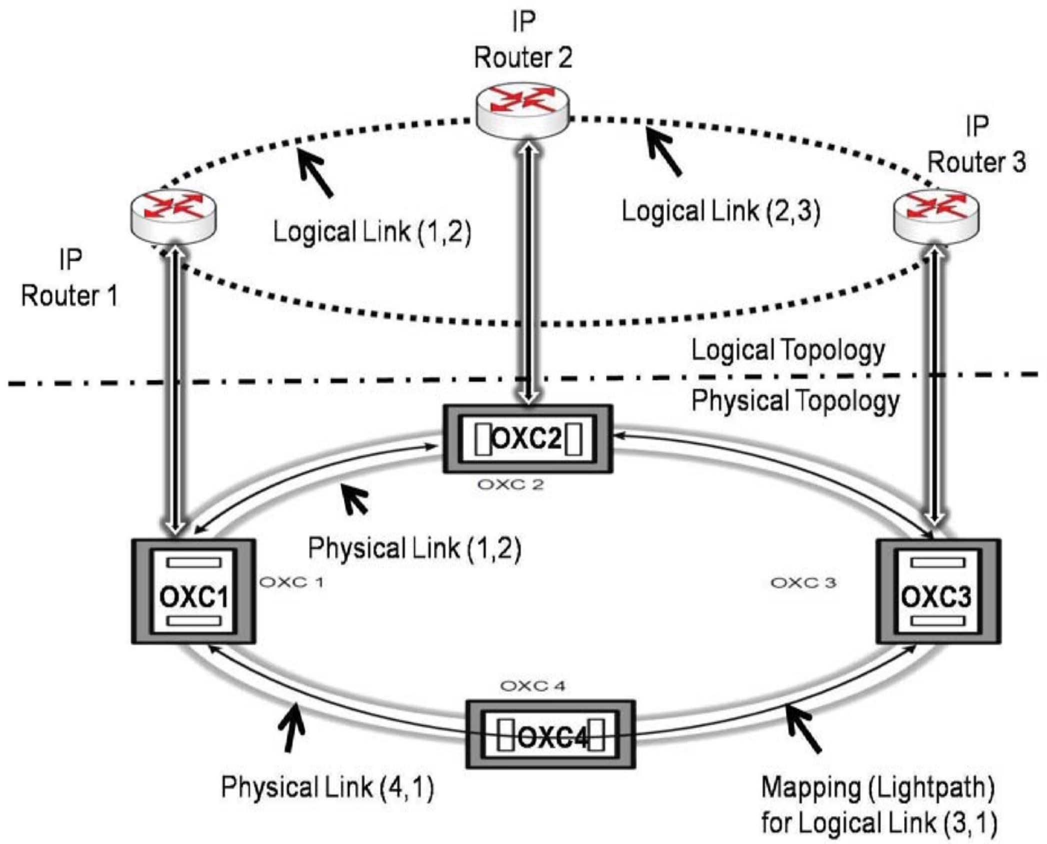


Figure 1.15: An IP-over-WDM network

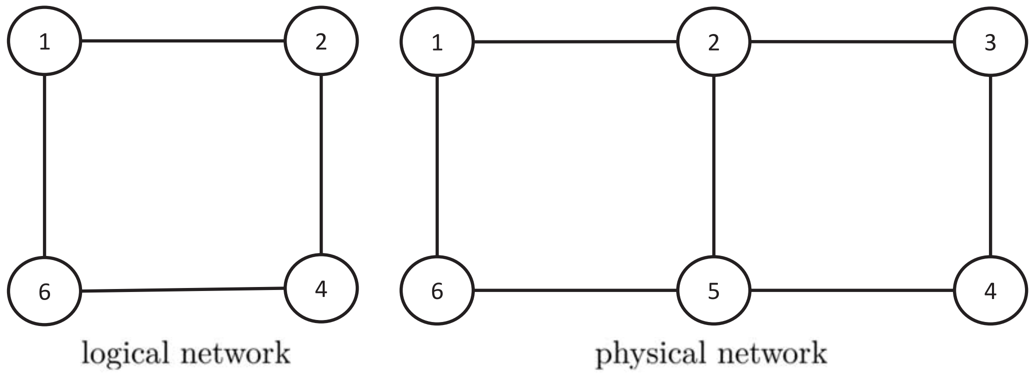
1.3.1 Survivable IP-over-WDM Networks

In this section, we discuss survivability issues in the IP-over-WDM network that use the physical layer protection and the logical layer restoration.

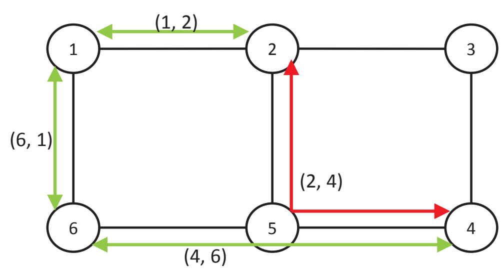
The reference [14] provides an overview of the above two possible fault-management techniques. For the physical layer protection, the protection mechanisms discussed in Section 1.2.3 provide methods to keep the survivability in the IP-over-WDM network. A backup lightpath for every primary lightpath is set in the physical network such that the single physical link failure does not disconnect both the primary and backup lightpaths and all traffic on the primary lightpath can be diverted to the backup lightpath. Two path protection mechanisms are considered: dedicated-path protection and shared-path protection. For the dedicated-path protection, a fiber-disjoint backup path and wavelengths are reserved for each primary path. The backup wavelength reserved on the links of backup path are not shared with other backup paths. For the shared-path protection, a fiber-disjoint backup path and wavelength are reserved. But the backup wavelengths are reserved on links of the backup path that may be shared with other backup paths. Hence, backup channels are shared with different backup paths among different failure scenarios.

For the logical layer restoration, an over-provision of the network is proposed such that after a physical link failure, the network still carries all the traffic it was carrying before. An *autonomous systems* (AS) consists of a set of routers that belong to the same administrative domain. When a link fails along a primary path between two nods/routers in AS, the *interior gateway protocol* (IGP) can dynamically find an alternative path between two nodes.

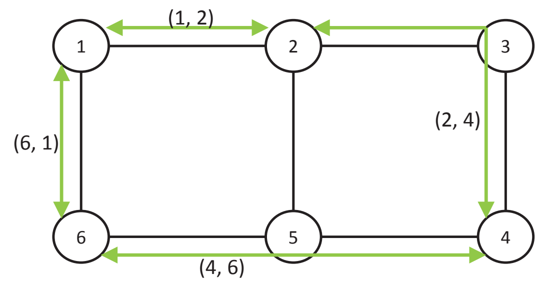
Modiano and Narula-Tam proposed in [5] a different mechanism for survivability in the IP-over-WDM networks. After each physical link failure, all lightpaths passing through this physical link are disconnected. The IP routers adjacent to the failed



(a) Logical and physical networks



(b) An unsurvivable routing



(c) A survivable routing

Figure 1.16: Unsurvivable and survivable mapping for logical topology

physical link can detect the failure and find alternative routes for failed lightpaths. Only if the logical links are mapped into the physical network in a way that the logical network retains connectivity after a single physical link failure, the survivable routing exists in the given IP-over-WDM network. Otherwise, some of the IP demands cannot be satisfied.

Examples of a survivable mapping and an unsurvivable mapping of the links of a logical topology onto the links of a physical topology are shown in Fig. 1.16. In the mapping of Fig. 1.16b, when physical link (4,5) fails, logical links (2,4) and (4,6), whose lightpaths are both routed through physical link (4,5), fail simultaneously causing the logical topology to become disconnected since logical node 4 is no longer connected to other nodes in the logical topology after this physical link failure. In contrast, in Fig. 1.16c no physical link failure can disconnect the logical topology, hence the mapping is survivable. Therefore, survivability of a mapping can be guaranteed if the lightpaths in the physical topology corresponding to this mapping are all link-disjoint.

1.4 Dissertation Organization and Contributions

This dissertation is organized as follows. In Chapter 2, we review the different approaches proposed in the literature to design survivable IP-over-WDM networks. In Chapter 3, we review different SMART-based algorithms that form the basis of most of the results in this dissertation.

In Chapter 4, we discuss how to augment the logical network with new logical links to guarantees survivability against multiple physical link failures. First, we show that if a logical topology is a chordal graph, then, it admits a survivable mapping as long as the physical topology is 3-edge connected and the logical topology is 2-edge connected. Second, we demonstrate how to embed a chordal graph on a logical topology to guarantee survivability under multiple physical link failures.

In Chapter 5, we first define the concept of robustness of a logical mapping algorithm which captures the ability of the algorithm to provide survivability against multiple physical failures. Robustness of an algorithm is the fraction of the cuts of logical network that are protected by the algorithm. Second, we analyze different algorithms for their robustness properties. We demonstrate that SMART-based algorithms guarantee high robustness under multiple physical link failures.

In Chapter 6, we define weakly and strongly survivable in a capacitated IP-over-WDM network. We provide exact MILP formulations and heuristics for the strongly and weakly survivable mappings in capacitated IP-over-WDM networks. We also consider the issue of spare capacity assignment at the physical layer to achieve strong survivability.

In Chapter 7, we outline directions for future research and summarize our research on survivable design in IP-over-WDM networks.

Chapter 2

Literature Review

An IP-over-WDM network implements IP directly over a WDM network by mapping a set of given IP connections as lightpaths in the WDM network [14][15]. A lightpath is an all-optical connection established by finding a path between the source and the destination of an IP connection in the WDM network and assigning it a wavelength [8]. Such networks use OXCs to switch network traffic (lightpaths) in the WDM layer and IP routers to route/reroute IP connections at the IP layer [14][15]. The set of IP routers and connections form the logical topology, and OXCs along with actual optical fibers form the physical topology. In the literature, it is common to refer to IP connections as IP or logical links, IP routers as logical nodes, OXCs as physical nodes and fibers connecting the OXCs as physical links.

An optical fiber simultaneously carries several lightpaths. Therefore, the failure of an optical fiber disconnects all the carried lightpaths, causing multiple failures in the logical topology, which can severely impact the entire network performance. Mechanisms that allow networks to deliver an acceptable level of service in the presence of a physical edge failure are referred to as survivability mechanisms and IP-over-WDM networks that implement such mechanisms are called survivable IP-over-WDM networks (henceforth, simply called survivable networks) [15]. Here, we only consider link-survivable networks i.e., networks that provide an acceptable level of service in the presence of one or more single link failures. The two widely discussed survivability mechanisms in the literature are protection and restoration [14][15]. Protection is generally provided at the physical layer but can be implemented at the logical layer also [14][15]. It requires a dedicated backup lightpath for each working lightpath such that the two lightpaths are link-disjoint. The backup path is used only when the working lightpath fails [15]. It is always possible to find two disjoint lightpaths if the physical topology is at least 2-edge-connected [16][17]. Restoration is usually provided at the logical layer by setting up working lightpaths for the IP connections and then provisioning the physical network with some additional (spare) capacity that is used by the IP routers to find backup lightpaths for the failed working lightpaths [14][15]. However, backup paths can be guaranteed only if the IP topology is initially embedded in such a way that it stays connected after a failure [5][18]. The references [5] and [18] established the necessary and sufficient conditions for an IP-over-WDM network employing restoration to be survivable. An IP-over-WDM network employing restoration is survivable under a single link failure only if none of the cutsets of the logical topology is carried by a single physical link. However, the fact that the number of cutsets in a network is exponential in the number of nodes makes the problem intractable [19].

The survivable logical topology design problem in IP-over-WDM networks has been widely studied in previous research. In the following sections, we first describe the problem and its settings and then we discuss the two major lines of pioneering works for this problem. One approach uses Integer Linear Programming formulations and the other is based on the duality between circuits and cutsets in a graph and provides a framework for a structural study of the problem.

2.1 Problem Description

Given a 2-edge-connected physical network $G_P = (V_P, E_P)$, where V_P is a set of physical nodes and E_P is a set of physical edges in G_P. A physical edge $e = (i, j) \in E_P$ connects a pair of terminal nodes $i, j \in V_P$, and is formed by bi-directional fiber links.

In other words, if there exists a link from node i to node j in G_P, then a link from node j to node i also exists. A non-negative value, c_e, associated with each physical link e is the maximum flow e can carry.

The logical topology $G_L = (V_L, E_L)$ is another network layer defined by a set of logical nodes $V_L \subseteq V_P$ and a logical edge set E_L, where E_L is a set of bi-directional logical links (s, t) between nodes s and t. A source-destination pair (s_i, t_i) is also referred to as a *commodity* i.

A *mapping* M_i of a logical link $(s_i, t_i) \in E_L$ in G_P is a physical path P_i in G_P which connects the corresponding physical nodes s_i, t_i in G_P. M_{E_L} is a set of mappings for all logical links $e \in E_L$. Such a mapping also represents the lightpath routing for logical links.

The survivable logical topology design problem is to find mappings for some or all logical links (s_i, t_i) such that the failure of any physical link $e \in E_P$ does not disconnect the logical network G_L. This is called a link-survivable mapping. For a simple logical ring structure, link survivability can be achieved by finding edge-disjoint mapping for each pair of logical links. In this dissertation we only consider single or multiple physical link failures.

The following section introduces the first line of research based on integer linear programming.

2.2 Integer Linear Programming Approach

In this approach a survivable logical topology design problem is formulated as an ILP. Once the ILP formulation is solved the results provide the exact solution for the problem. The main drawback of this approach is that an exponential number of constraints, involving all cutsets in a logical network, are to be satisfied. Modiano and Narula-Tam [5] first provided the necessary and sufficient conditions for a survivable logical topology routing and also proposed an ILP formulation to solve it. Their approach is detailed below.

- *Cutset Based Method:* Given a partition $(S, V_L \setminus S)$ of the node set V_L, the set of edges with one node in S and the other in $V_L \setminus S$ is called a cutset. This cutset is denoted by $CS(S, V_L \setminus S)$. Most ILP approaches applied to solve the survivable logical topology design problem are based on cutsets, which was initiated in [5]. Modiano and Narula-Tam formally showed in [5] that the problem of finding survivable mappings is NP-complete for general and ring logical topologies. Therefore, they provided ILPs to find a solution. The ILPs are based on the observation that a logical topology can be disconnected after the failure of a physical link only if the failed physical link carries the lightpaths of the set of all logical links belonging to a cutset of the logical topology. Alternatively, every cutset of the logical topology must contain at least a pair of edges whose lightpaths are pair-wise disjoint in order for the mappings to be survivable. Now we briefly review their main results.

Definition 2.1 [5] *A routing is* survivable*, if the failure of any physical link leaves the logical network connected.*

Theorem 2.1 [5] *A routing is survivable if and only if for every cutset $CS(S, V_L \setminus S)$ of the logical topology the following holds. Let $E(s,t)$ be the set of physical links used by logical link (s,t). Then, for every cutset $CS(S, V_L \setminus S)$,*

$$\bigcup_{(s,t) \in CS(S, V_L \setminus S)} E(s,t) = \emptyset.$$

Based on Theorem 2.1, the following integer programming formulation for the survivable IP-over-WDM network design is proposed in [5].

(a) *Flow conservation constraints:*

$$\sum_{(i,j)\in E_P} f_{ij}^{st} - \sum_{(j,i)\in E_P} f_{ji}^{st} = 1 \qquad \text{if } s = i, (s,t) \in E_L$$

$$\sum_{(i,j)\in E_P} f_{ij}^{st} - \sum_{(j,i)\in E_P} f_{ji}^{st} = -1 \qquad \text{if } t = i, (s,t) \in E_L$$

$$\sum_{(i,j)\in E_P} f_{ij}^{st} - \sum_{(j,i)\in E_P} f_{ji}^{st} = 0 \qquad \text{otherwise}$$

(b) *Survivability constraints:*

$$\sum_{(s,t)\in CS(S,V_L\setminus S)} f_{ij}^{st} + f_{ji}^{st} \leq |CS(S, V_L \setminus S)|, \qquad S \in V_L, (i,j) \in E_P \tag{2.1}$$

(c) *Objective function:*

$$\min \sum_{(i,j)\in E_P,(s,t)\in E_L} f_{ij}^{st}$$

where f_{st}^{ij} is a binary variable which represents whether (s,t) is routed through (i,j) with $(i,j) \in E_P$, $(s,t) \in E_L$. Constraint (2.1) is the cutset based constraint, which follows from Theorem 2.1.

However, the ILP does not scale well as it must examine all possible cuts, a number that grows exponentially with the size of the topology.

Based on [5] Todimala and Ramamurthy [20] [19] studied the problem with *Shared Risk Link Group* (SRLG), R_p, on physical links and provided improved ILPs. Their main conclusion is as follows.

Definition 2.2 [19] *A cut, $(S, V_L \setminus S), S \subseteq V_L$, is called a primary-cut (denoted by $P_{G_L}(S, V_L \setminus S)$) if and only if both of the induced sub-graphs of a network G_L by the node set S and $V_L \setminus S$ are connected components.*

Theorem 2.2 [19] *Given the physical topology G_P, SRLG set R_p and the logical topology G_L, the routing of G_L over G_P is survivable if and only if for any $r \in R_p$ and for all primary cutsets $P_{G_L}(S, V_L \setminus S)$ of the logical topology at least one link in $P_{G_L}(S, V_L \setminus S)$ is not routed over any links in r.*

Let $g_r^{st} = 1$ if the logical link (s,t) is routed over at least one of the physical links that belongs to the SRLG set r. Based on Theorem 2.2, Todimala and Ramamurthy [19] proposed the following cutset based survivable constraints.

(a) *Decision on whether a logical edge is routed through SRLG*

$$g_r^{st} \leq \sum_{(i,j)\in r} f_{ij}^{st}, \qquad r \in R_p, (s,t) \in E_L$$

$$|E_P| g_r^{st} \geq \sum_{(i,j)\in r} f_{ij}^{st}, \qquad r \in R_p, (s,t) \in E_L$$

(b) *SRLG survivable constraint*

$$\sum_{(s,t)\in P_{G_L}(S,V_L\setminus S)} g_r^{st} < |CS(S, V_L \setminus S)|, \qquad r \in R_p, P_{G_L}(S, V_L \setminus S) \in G_L$$

The proposed ILP incorporates wavelength assignment constraints and only considers primary cuts, but does not scale well either. However, when applied to planar cycles and hierarchical planar cycles, the ILP can be solved fairly quickly [19].

In reference [21], certain metrics are defined that capture the quality of a lightpath routing. Specifically, the concept of *Minimum Cross Layer Cut* (MCLC) is defined in [21]. This metric is a measure of the ability of a routing to tolerate multiple physical edge failures. Finding a routing that maximizes MCLC is also intractable. An ILP formulation to find a survivable routing that maximizes a measure that is related to MCLC is given in [21]. Kan et al. in [22] discussed the relationship between survivable lightpath routing and the spare capacity requirements on the logical links to satisfy the original traffic demands after failures. ILP formulations to find a routing that minimizes spare capacity requirements is presented. Lee et al. in [23] incorporated the link failure probabilities to describe survivability in logical topology. In a recent work [24], we considered the general case of capacitated IP-over-WDM networks with capacities on physical edges and demands on logical edges.

Different from Modiano's approach, Deng et al. in [25] proposed an alternative MILP formulation which does not require enumeration of all cutset of the topology. The approach in [25] utilized the idea that a mapping for the logical topology is survivable if after any physical link failure, there exists at least one spanning tree in the logical topology.

All the works based on ILPs mentioned above have the common drawback that the computation cost grows exponentially due to the number of constraints, which make these approaches infeasible when the networks to be considered have size larger than a few dozen nodes. Hence approximation algorithms or heuristics are usually proposed in order to provide solutions close to the optimal in a timely manner.

2.3 Structural Approach

The structural approach to the cross-layer survivability was initiated in [26] and was later extended and generalized in [27] and [28].

- *Circuit Based Method – SMART Algorithm*: The ILP solutions can provide a survivable routing when there exists a feasible solution to the formulations. However, the exponential number of cutsets in a graph is a bottleneck for the algorithm. To overcome this problem, Kurant and Thiran in [26] provided a framework called SMART and introduced the concept of *piecewise survivability*. Instead of exhaustively searching for all the cutsets in a graph, SMART utilizes circuits to find survivable mappings for logical topologies. The framework repeatedly picks connected pieces (subgraphs) of the logical topology and finds survivable mappings for these pieces. If a survivable mapping is found for a piece, its links are short-circuited (contracted) and the algorithm proceeds by picking another piece. The process is repeated until the logical topology is reduced to a single node or a search for a piece with survivable mapping is unsuccessful. If the logical topology is reduced to a single node, a survivable mapping for the logical topology has been found; otherwise a survivable mapping does not exist.

Figure 2.1 shows the contraction of a network. To contract links $\{a, b, c\}$, links $\{a, b, c\}$ and their adjacent nodes are removed and a new node is inserted to replace the removed nodes. After the contraction, the remaining links d, e, f, i in G connect uncontracted nodes to the contracted node.

The SMART approach is actually based on the following theorem.

Definition 2.3 [29] *An open ear decomposition of an undirected graph is a partition of the edges into a simple cycle P_0 and simple paths $P_1, P_2, \ldots, P_k$ such that for each $i > 0$, P_i is joined to previous paths only at its (2 distinct) ends, i.e., $V(P_i) \bigcap V(\bigcup_{j<i} P_j)$ consists of the 2 ends of P_i. $P_0, P_1, \ldots P_k$ are called ears of the decomposition.*

Theorem 2.3 [17] *An undirected graph is bi-connected if and only if it has an open ear decomposition.*

Based on Theorem 2.3, if the logical topology is two-connected, it can be decomposed into ears. An ear can be treated as a cycle in the contracted graph where some nodes in the cycle may represent a group of contracted nodes. Figure 2.1 shows that if we find the survivable mapping for all the edges in $A = \{a, b, c\}$, we can contract the nodes in A and substitute the contracted nodes with a new contracted node, where the contracted node and an ear, $\{e, f\}$ for example, form a cycle.

The SMART algorithm is described in Fig. 2.2.

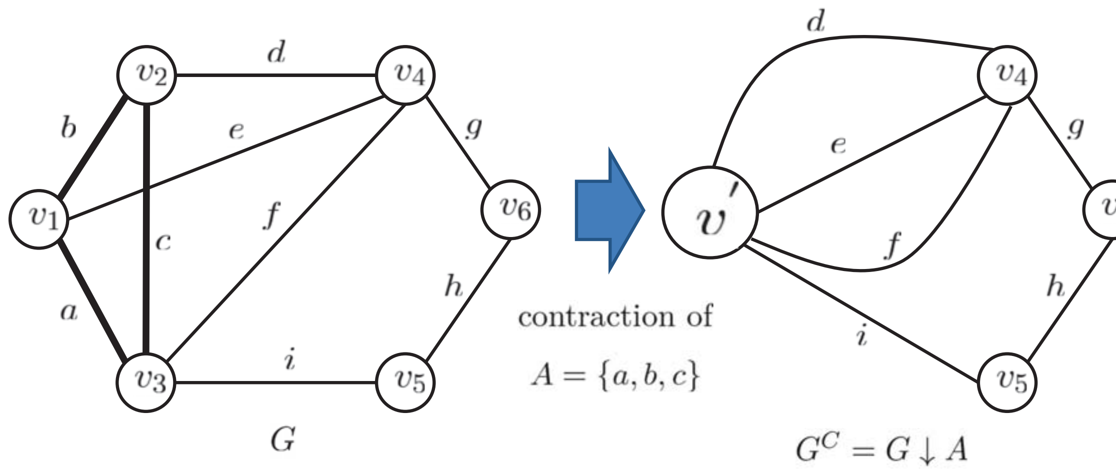


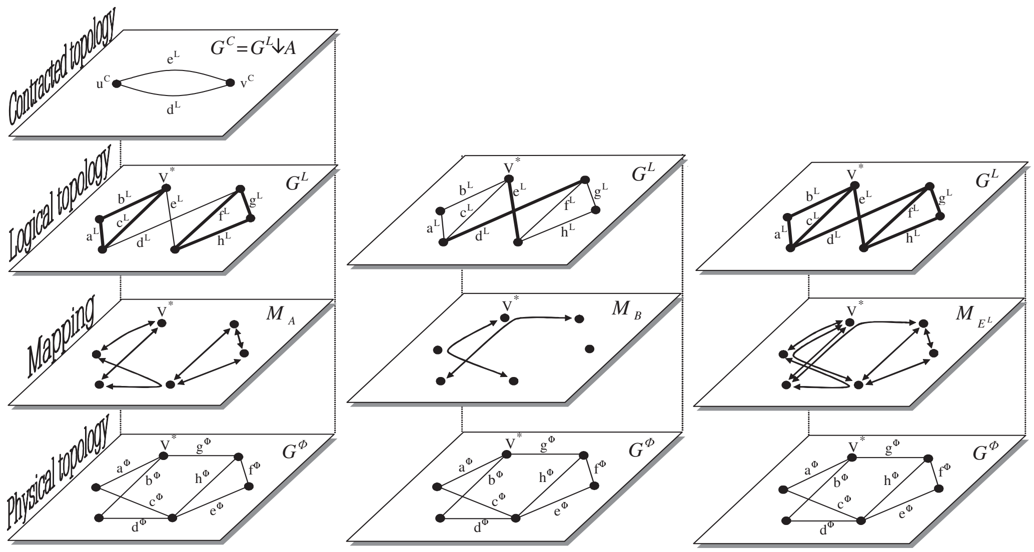
Figure 2.1: Graph contraction

Input: Physical topology G^P, logical topology G^L, an empty mapping $M_A = \emptyset$
Output: M_A

1: Assign the contracted topology $G^C = G^L$
2: Take a connected subgraph $G^C_{sub} = (V^C_{sub}, B)$ of the contracted topology G^C and find a mapping M_B of links B in G^C_{sub} such that M_B is a link-survivable mapping on G^C_{sub}. If M_B does not exist for any subgraph G^C_{sub}, return M_A and terminate the algorithm.
3: Update the mapping by merging M_A and M_B, $M_A = M_A \bigcup M_B$.
4: Contract G^C on B. ($G^C = G^C \downarrow B$; $\downarrow$: *contraction*).
5: If G^C is a single node, return M_A. End the algorithm.
6: Goto 2.

Figure 2.2: SMART algorithm

The implementation in [26] takes a cycle as the subgraph in step 2 since cycles are the most basic 2-connected graphs. Finding a survivable mapping for a cycle is equivalent to finding mutually disjoint paths in the physical layer, which is an NP-complete problem [30]. Hence Kurant and Thiran in [26] apply a simple heuristic by first assigning all the edges in the physical layer with an initial cost 1 and then finding the shortest paths in the physical layer for each logical link. If there are no common edges on the paths, the paths form the survivable routing for the subgraph. Otherwise, SMART algorithm increases the cost of the overlapped edge by 1 and



(a) (b) (c)

Figure 2.3: The mapping examples from the logical layer to the physical layer

continues the algorithm. This heuristic stops after a certain number of unsuccessful iterations.

Figure 2.3 [26] illustrates the step of contraction with respect to the logical and physical topologies and presents the contracted topology. Two connected subgraphs $\{a^L, b^L, c^L\}$ and $\{f^L, g^L, h^L\}$ are shown in Fig. 2.3a. SMART algorithm first selects the $\{a^L, b^L, c^L\}$ subgraph and finds the edge-disjoint mapping $\{\{c^\Phi, d^\Phi\}, a^\Phi, b^\Phi\}$. The subgraph $\{a^L, b^L, c^L\}$ is then contracted to a node u^c. Next the $\{f^L, g^L, h^L\}$ subgraph is mapped to $\{h^\Phi, f^\Phi, e^\Phi\}$ and contracted to v^c in the following step. After the two contractions the original logical topology becomes the contracted topology on top of Fig. 2.3a. Figure 2.3b shows the contraction of the edges e^L, d^L on the contracted topology and Fig. 2.3c gives the final survivable routing.

Compared to the algorithm in [19], SMART algorithm can find survivable routing for not only a ring but the whole logical topology. On the other hand, when SMART encounters early termination because of several unsuccessful trials of finding survivable mapping, it still provides piecewise survivable mappings. Unlike the ILP solutions which do not provide the information for further processing, the SMART algorithm can provide insight into the possibility of augmenting the logical links so that the augmented graph becomes survivable.

While SMART algorithm in [26] is applicable to large networks, it cannot deal with multiple failures. Kurant and Thiran [31] extended the SMART algorithm to handle multiple link failures through finding the $k - survivable$ mapping for the subgraphs selected in step 2 of the SMART algorithm.

Thulasiraman et al. [32][27][28], Javed et al. [33] [34] [35] addressed several issues related to the SMART approach. In references [32][27][28] several structural issues related to the SMART approach have been studied using the concept of duality between circuits and cutsets. These will be elaborated in the following chapter.

Javed et al. [34] use a concept of randomized routing introduced in [36] to help find edge-disjoint paths. Javed et al. [35] also proposed heuristics to combine the protection and restoration mechanism (hybrid) where fewer protection capacity is required for survivable routing.

In our recent works [32] [37] [38], we proposed approaches to augment a logical topology with additional links so that the augmented topology becomes survivable, and we also studied the robustness of SMART-based algorithms for multiple failures.

2.4 Other Related Works

Rather than evaluating all the cutsets, Ducatelle and Gambardella [39] employed a probability function as an estimate of the cutsets and iteratively solved the problem with local search algorithm.

Crochat et al. [40] provided a comprehensive framework for the logical topology mapping problem in IP-over-WDM networks and defined three constraints (including survivability) that a solution must satisfy. They showed that the problem is

NP-complete and suggested a heuristic algorithm based on Tabu search. Stern and Bala [6] suggested a hybrid approach to survivability that uses a combination of restoration and protection.

Chapter 3

SMART-based Algorithms

In this chapter, we present all SMART-based algorithms introduced in [27][28] in detail, which will form the basis of works presented in the following chapters.

Duality between circuits and cuts in a graph is one of the well-studied topics in graph theory. This concept has played a significant role in the development of methodologies for solving problems in various applications. Most of the early results in electrical circuit theory were founded on the duality relationship between circuits and cuts [41]. There is a wealth of literature on the role of duality in network optimization (that is, discrete optimization on graphs and networks) [42]. Most often, for a primal algorithm based on circuits there is a dual algorithm based on cuts for the same problem. The primal and dual algorithms possess certain characteristics that make one superior to the other depending on the application. SMART algorithm for the survivable logical topology mapping problem is based on circuits. The question then arises whether there exists a dual methodology based on cuts. The work in [28] answered this question in the affirmative and provided a unified algorithmic framework for the survivable logical topology mapping (SLTM) problem. Thulasiraman et al. [28] also provided much insight into the structure of solutions for the SLTM problem. We discuss this framework and corresponding algorithms in the following section. We also present without proofs certain results from the graph theory literature that will be of interest in our developments for the following sections.

3.1 Survivable Logical Topology Mapping Problem and a Unified Algorithmic Framework

Given a spanning tree T with branches $\{b_1, b_2, \ldots, b_{(n-1)}\}$ and chords $\{c_1, c_2, \ldots, c_{(m-n+1)}\}$ of a graph with n nodes and m edges, the *fundamental circuit matrix* $B_f = [b_{ij}]_{(m-n+1)\times(m)}$ has one row for each chord/fundamental circuit and one column for each edge. With $B(c_i)$ denoting the row corresponding to chord c_i the entry b_{ij} is defined as

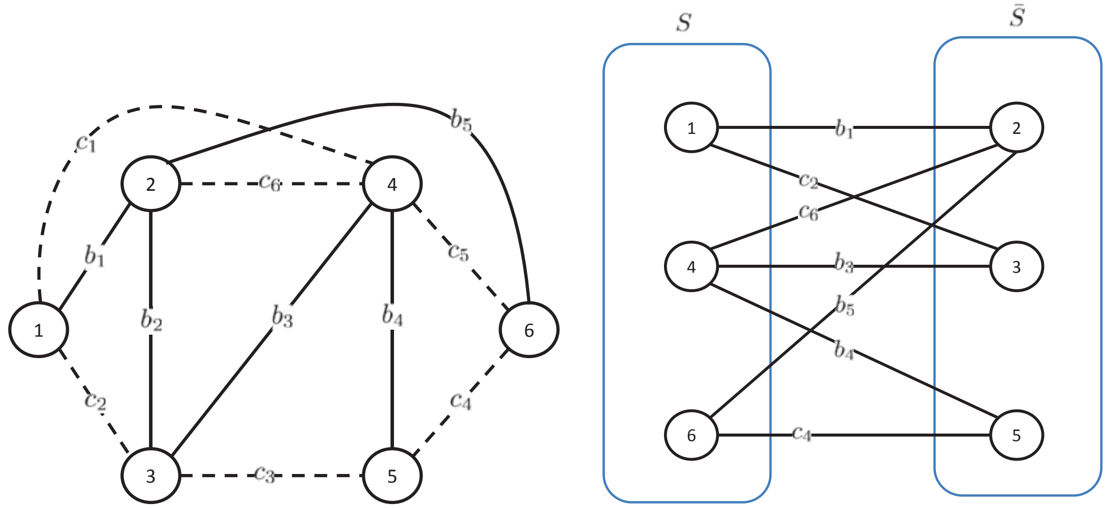
$$
\begin{aligned}
b_{ij} &= 1 \text{ if } B(c_i) \text{ contains edge } j, \\
&= 0 \text{ otherwise.}
\end{aligned}
$$

Arranging the rows of B_f such that the j^{th} row ($j \leq m-n+1$) corresponds to the fundamental circuit $B(c_j)$ and arranging the columns in the order $\{c_1, c_2, \ldots, c_{(m-n+1)}, b_1, b_2, \ldots, b_{(n-1)}\}$ we can write the B_f matrix as $B_f = [\bigcup | B_{ft}]$, where $\bigcup$ is the unit matrix of size $(m-n+1)$. For example, the B_f matrix with respect to the spanning tree T of Fig. 3.1a is given in (3.1).

In a similar manner the *fundamental cutset matrix* with respect to the tree T can be defined as $Q_f = [q_{ij}]_{(n-1)\times(m)}$. Q_f has $(n-1)$ rows, one for each branch/fundamental cutset and one column for each edge. With $Q(b_i)$ denoting the row corresponding to branch b_i the entry q_{ij} is defined as

$$
\begin{aligned}
q_{ij} &= 1 \text{ if } Q(b_i) \text{ contains edge } j, \\
&= 0 \text{ otherwise.}
\end{aligned}
$$

Arranging the rows of Q_f such that the j^{th} row corresponds to f-cutset $Q(b_j)$ and the columns correspond to edges in the order $\{b_1, b_2, \ldots, b_{n-1}, c_1, c_2, \ldots, c_{(m-n+1)}\}$ the



(a) Graph G (b) Cut S

Figure 3.1: (a) A graph with a spanning tree (bold lines); (b) A cut

Q_f matrix can be written as $Q_f = [\bigcup | Q_{fc}]$. For example, the Q_f matrix with respect to the tree T of Fig. 3.1a is given in (3.2).

$$\begin{array}{c|cccccc|ccccc|}
 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & b_1 & b_2 & b_3 & b_4 & b_5 \\
\hline
c_1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
c_2 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
c_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
c_4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
c_5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
c_6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0
\end{array} \tag{3.1}$$

$$
\begin{array}{c} \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{array}
\begin{array}{c}
\begin{array}{ccccc|cccccc} b_1 & b_2 & b_3 & b_4 & b_5 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array} \\
\left[\begin{array}{ccccc|cccccc}
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0
\end{array}\right]
\end{array}
\tag{3.2}
$$

In the following definitions, $B(c_i)$ and $Q(b_i)$ will also be used to denote the sets of edges in the corresponding fundamental circuit and fundamental cutset, respectively. An ordered sequence $B(c_1), B(c_2), \ldots, B(c_k)$ is a *circuit cover sequence* or simply a *B-sequence* of length k if

- $[B(c_j) - c_j - \bigcup_{p=1}^{j-1} B(c_p)] \neq \emptyset, 2 \leq j \leq k$,
- $\bigcup_{p=1}^{k} B(c_p) = E - \{\text{chords not in the B-sequence}\}$.

Note that for a given spanning tree and its f-circuits, there may be more than one B-sequence. For example for the fundamental circuits given in (3.1), following are the three B-sequences:

- $B(c_1), B(c_3), B(c_5)$,
- $B(c_4), B(c_1)$,
- $B(c_6), B(c_1), B(c_4)$.

Note that the order in which the $B(c_j)$'s appear matters in the definition of B-sequences. Without loss of generality assume that $B(c_1), B(c_2), \ldots, B(c_k)$ is a B-sequence of length k. Let us define $S(c_j)$ as follows:

- $S(c_1) = B(c_1) - c_1$,
- $S(c_j) = B(c_j) - c_j - \bigcup_{p=1}^{j-1} B(c_p), 2 \leq j \leq k$.

Then the submatrix of the f-circuit comprised of the rows corresponding to $B(c_1), B(c_2), \ldots, B(c_k)$ will have the structure shown in (3.3). Note that $\times$ represents either 0 or 1 in (3.3). Let the set of chords not in the circuit cover sequence be called *unmapped chords*.

$$
\begin{array}{c|cccccc|ccc|ccc|c|cc|c|cc|}
 & c_1 & c_2 & \cdots & c_j & \cdots & c_k & & S(c_1) & & & S(c_2) & & \cdots & S(c_j) & & \cdots & S(c_k) & \\
\hline
c_1 & 1 & 0 & \cdots & 0 & \cdots & 0 & 1 & 1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\
c_2 & 0 & 1 & \cdots & 0 & \cdots & 0 & \times & \times & \times & 1 & 1 & 1 & \cdots & 0 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots \\
c_j & 0 & 0 & \cdots & 1 & \cdots & 0 & \times & \times & \times & \times & \times & \times & \cdots & 1 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots \\
c_k & 0 & 0 & \cdots & 0 & \cdots & 1 & \times & \times & \times & \times & \times & \times & \cdots & \times & \times & \cdots & 1 & 1
\end{array}
\tag{3.3}
$$

An ordered sequence $Q(b_1), Q(b_2), \ldots, Q(b_k)$ is a *cutset cover sequence* or simply a *Q-sequence* of length k if

- $[Q(b_j) - b_j - \bigcup_{p=1}^{j-1} Q(b_p)] \neq \emptyset, 2 \leq j \leq k,$
- $\bigcup_{p=1}^{k} Q(b_p) = E - \{\text{branches not in the } Q - \text{sequence}\}.$

Note that for a given spanning tree and its f-cutsets, there may be more than one Q-sequence. For example for the fundamental cutsets given in (3.2), following are the three Q-sequences.

- $Q(b_4), Q(b_5), Q(b_2),$
- $Q(b_4), Q(b_5), Q(b_1), Q(b_2),$
- $Q(b_1), Q(b_2), Q(b_4).$

Without the loss of generality assume that $Q(b_1), Q(b_2), \ldots, Q(b_k)$ is a Q-sequence of length k. Let us define $\hat{S}(b_j)$ as follows:

- $\hat{S}(b_1) = Q(b_1) - b_1$,
- $\hat{S}(b_j) = Q(b_j) - b_j - \bigcup_{p=1}^{j-1} Q(b_p), 2 \leq j \leq k$.

Then the submatrix of the f-cutset comprised of the rows corresponding to $Q(b_1), Q(b_2), \ldots, Q(b_k)$ has a structure similar to (3.3) as shown in (3.4). Let the set of branches not in the cutset cover sequence be called *unmapped branches*.

$$
\begin{array}{c}
\begin{array}{cccccc|ccc|ccc|c|cc|c|cc}
 & b_1 & b_2 & \cdots & b_j & \cdots & b_k & & \hat{S}(b_1) & & & \hat{S}(b_2) & & \cdots & \hat{S}(b_j) & & \cdots & \hat{S}(b_k) &
\end{array} \\
\begin{array}{c}
b_1 \\ b_2 \\ \vdots \\ b_j \\ \vdots \\ b_k
\end{array}
\left[
\begin{array}{cccccc|ccc|ccc|c|cc|c|cc}
1 & 0 & \cdots & 0 & \cdots & 0 & 1 & 1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\
0 & 1 & \cdots & 0 & \cdots & 0 & \times & \times & \times & 1 & 1 & 1 & \cdots & 0 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots \\
0 & 0 & \cdots & 1 & \cdots & 0 & \times & \times & \times & \times & \times & \times & \cdots & 1 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & \cdots & 1 & \times & \times & \times & \times & \times & \times & \cdots & \times & \times & \cdots & 1 & 1
\end{array}
\right]
\end{array}
\tag{3.4}
$$

We now summarize a few standard results that will be of interest in the developments in the following sections.

Theorem 3.1 (a) *If a cut contains the branches* $\{b_1, b_2, \ldots, b_j\}$ *then the corresponding cut vector can be represented as modulo 2 addition of the vectors* $Q(b_1), Q(b_2), \ldots, Q(b_j)$. *That is, the cut vector is equal to* $Q(b_1) \bigoplus Q(b_2) \bigoplus \ldots \bigoplus Q(b_j)$.
(b) *If a circuit contains the chords* $\{c_1, c_2, \ldots, c_j\}$ *then the corresponding circuit vector can be represented as* modulo 2 *addition of the vectors* $B(c_1), B(c_2), \ldots, B(c_j)$. *That is, the circuit vector is equal to* $B(c_1) \bigoplus B(c_2) \bigoplus \ldots \bigoplus B(c_j)$.

Theorem 3.2 (Orthogonality) *A circuit and a cut have an even number of common edges.*

Theorem 3.3 $B_{ft} = Q_{fc}^t$, *where* Q_{fc}^t *is the transpose of* Q_{fc}.

Theorem 3.4 (a) *Given a B-sequence* $B(c_1), B(c_2), \ldots, B(c_k)$, *let* $B(c_{i_1}), B(c_{i_2}), \ldots, B(c_{i_\ell})$ *be a sub-sequence of this sequence then* $S(c_{i_\ell}) \subseteq B(c_{i_1}) \bigoplus B(c_{i_2}) \bigoplus \ldots \bigoplus B(c_{i_\ell})$. (b) *Given a Q-sequence* $Q(b_1), Q(b_2), \ldots, Q(b_k)$, *let* $Q(b_{i_1}), Q(b_{i_2}), \ldots, Q(b_{i_k})$ *be a subsequence of this sequence then* $Q(c_{i_\ell}) \subseteq \hat{S}(c_{i_1}) \bigoplus \hat{S}(c_{i_2}) \bigoplus \ldots \bigoplus \hat{S}(c_{i_\ell})$.

Theorem 3.5 [41] *A graph is connected if and only if every cut of the graph contains at least one edge.*

Using the above concepts four algorithms CIRCUIT-SMART (Fig. 3.2), CUTSET-SMART (Fig. 3.3), CUTSET-SMART-SIMPLIFIED (Fig. 3.4) and INCIDENCE-SMART (Fig. 3.5) were proposed in [27]. To guarantee survivability, these algorithms add to the logical graph new edges in parallel to some of the edges whenever necessary. These edges will be called *protection edges*. The input to these algorithms is a physical topology G_P and a logical topology G_L. The output of these algorithms is a survivable logical graph $G_L^{'}$ containing G_L.

Theorem 3.6 [27] *Algorithm INCIDENCE-SMART provides a survivable mapping of the edges of a logical graph* G_L.

Proof: We first define a logical node with the maximum degree as the datum vertex. Let $v_1, v_2, \ldots, v_{n-1}$ be the order in which the vertices have been considered by algorithm INCIDENCE-SMART. Consider any cut $(S, \overline{S})$ in G_L. Let the datum vertex be in $\overline{S}$. Let v_i be the vertex in S with the highest index. Then, in the current graph at the step when v_i is considered by the algorithm it will not be adjacent to any vertex in S. So, according to the algorithm v_i will be connected to at least two vertices in $\overline{S}$, and the corresponding edges connecting S and $\overline{S}$ are mapped into disjoint lightpaths, guaranteeing that at least one of these edges will remain in the cut after a single edge failure in the physical topology and so satisfying the condition of Theorem 3.5.

Since this is true for all cuts, the mapping generated by the algorithm is survivable. □

Input: A 2-edge connected physical topology G_P, a 2-edge connected logical topology G_L, a spanning tree T of G_L, a set of fundamental circuits and a B-sequence $B(c_1), B(c_2), \ldots, B(c_k)$.
Output: A survivable logical graph $G_L^{'}$ containing G_L.
1: **for** $i = 1, 2, \ldots, k$ **do**
2: Map a maximum subset of edges in $S(c_i) \bigcup c_i$ into disjoint lightpaths in G_P.
3: To all other edges in $S(c_i) \bigcup c_i$ add protection edges and map each edge and its protection edge into disjoint lightpaths in G_P. [30]
4: **end for**
5: Map all the chords not in the B-sequence into lightpaths in G_P arbitrarily.

Figure 3.2: Algorithm CIRCUIT-SMART

Input: A 2-edge connected physical topology G_P, a 2-edge connected logical topology G_L, a spanning tree T of G_L, a set of fundamental cutsets and a Q-sequence $Q(b_1), Q(b_2), \ldots, Q(b_k)$.
Output: A survivable logical graph $G_L^{'}$ containing G_L.
1: **for** $i = 1, 2, \ldots, k$ **do**
2: Map a maximum subset of edges in $\hat{S}(b_i) \bigcup b_i$ into disjoint lightpaths in G_P.
3: To all other edges in $\hat{S}(b_i) \bigcup b_i$ add protection edges and map each edge and its protection edge into disjoint lightpaths in G_P [30].
4: **end for**
5: To each unmatched branch b (branches not in $Q - sequence$), add a protection edge $b^{'}$ and map them into disjoint lightpaths in G_P.

Figure 3.3: Algorithm CUTSET-SMART

We first draw attention to a shortcoming of the algorithmic framework CUTSET-SMART. Algorithm CIRCUIT-SMART of [27] would not require any additional edges to be added to the logical graph if no new edges (protection edges) are added in step 3 of this algorithm. This is not the case with algorithm CUTSET-SMART. This algorithm requires protection edges to be added to all unmapped branches. So, CIRCUIT-SMART guarantees a survivable mapping of the given logical graph, if step 3 does not require any new edges to be added. On the other hand, CUTSET-SMART

Input: A 2-edge connected physical topology G_P, a 2-edge connected logical topology G_L, a spanning tree T of G_L, a set of fundamental cutsets and a Q-sequence $Q(b_1), Q(b_2), \ldots, Q(b_k)$.
Output: A survivable logical graph $G_L^{'}$ containing G_L.

1: **for** $i = 1, 2, \ldots, k$ **do**
2: Map b_i in disjoint manner with some cord in set $\hat{S}(b_i)$.
3: If this is not possible for any chord in $\hat{S}(b_i)$ then add a protection edge to one of these chords and map the chord and the protection edge in disjoint manner.
4: **end for**
5: To each unmatched branch b, add a protection edge $b^{'}$ and map them as disjoint lightpaths in G_P.
6: Map all the unmapped logical edges arbitrarily.

Figure 3.4: Algorithm CUTSET-SMART-SIMPLIFIED

Input: A 2-edge connected physical topology G_P, a 2-edge connected logical topology G_L and $INC-sequence\ INC(v_1), INC(v_2), \ldots, INC(v_k)$.
Output: A survivable logical graph $G_L^{'}$ containing G_L.

1: **for** $i = 1, 2, \ldots, k$ **do**
2: If vertex v_i has degree greater than or equal to 2 in the current graph, then map all the edges incident on v_i into disjoint lightpaths in G_P.
3: If the degree of v_i in the current graph is one, then add a new logical edge connecting v_i to the datum vertex. Then map this new edge and the only edge incident on v_i into disjoint lightpaths.
4: If degree of v_i in the current graph is zero add two new parallel logical edges connecting v_i to the datum vertex. Then map these two edges into disjoint lightpaths in G_P.
5: **end for**

Figure 3.5: Algorithm INCIDENCE-SMART

guarantees a survivable mapping of the graph obtained by contracting the unmapped branches in the logical graph, if step 3 of this algorithm does not require any new edges to be added. This issue was studied in [28] and a solution was also provided. We outline this work in Chapter 5.

Chapter 4

Logical Topology Augmentation for Guaranteed Survivability under Multiple Failures in IP-over-WDM Optical Networks

4.1 Introduction

The survivable logical topology mapping problem in an IP-over-WDM optical network is to map each link (u, v) in the logical topology into a lightpath between the nodes u and v in the physical topology such that failure of a single physical link does not cause the logical topology to become disconnected. Kurant and Thiran [43] presented an algorithmic framework called SMART that involves successive contracting of circuits in the logical topology and mapping the logical links in the circuits into edge- disjoint lightpaths in the physical topology. In a recent work [27] a dual framework involving cutsets was presented and it was shown that both these frameworks possess the same algorithmic structure. Algorithms CIRCUIT-SMART, CUTSET-SMART and INCIDENCE-SMART were also presented in [27]. All these algorithms suffer from one important shortcoming, namely, disjoint lightpaths for certain groups of logical links may not exist in the physical topology. Therefore, in such cases, we will have to augment the logical topology with new logical links to guarantee survivability. In this chapter we address this augmentation problem. We first identify a logical topology that admits a survivable mapping under a physical link failure as long as the physical topology is 3-edge-connected. We show how to embed this logical topology on a given logical topology so that the augmented topology admits a survivability mapping as long as the physical topology is 3-edge-connected. We then generalize these results to achieve augmentation for survivability of a given logical topology under multiple physical link failures. Finally, we define the concept of survivability index of a mapping. We provide simulation results to demonstrate that even when certain requirements of the generalized augmentation procedure are relaxed, our approach will result in mappings that achieve high survivability index.

4.2 A Survivable Logical Topology Structure and Augmentation for Single Link Failure Survivability

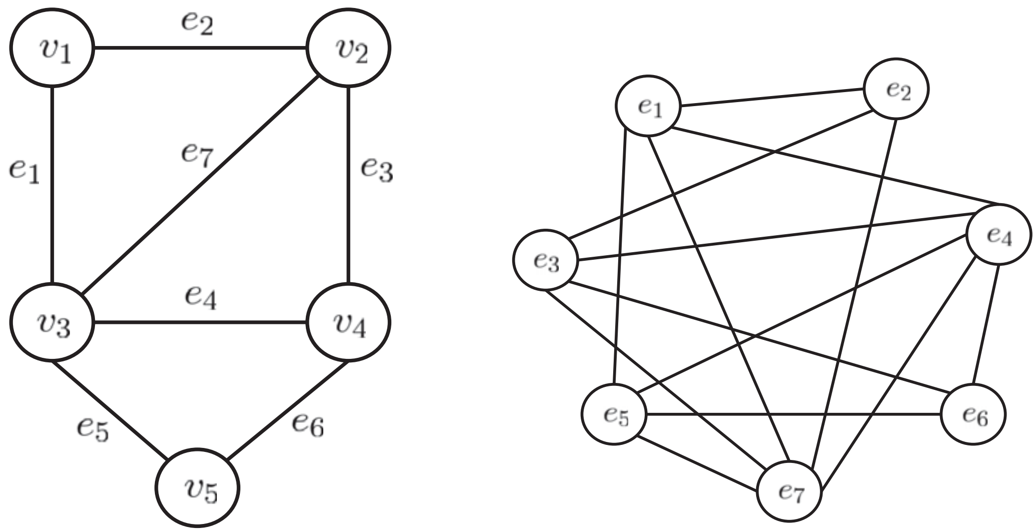
In this section, we first present a logical topology that always has a survivable mapping as long as the physical topology is 3-edge-connected. Recall that a protection edge is an edge in parallel to an existing edge. We then show how this topology can be used to augment any logical topology to guarantee a survivable mapping of the augmented topology.

We define a graph to be k-vertex-connected graph if at least k vertices have to be removed to disconnect the graph. We define the *line graph* of a graph as follows [44].

Given a graph G with m edges and n vertices, the line graph $L(G)$ of G has m vertices, with each vertex corresponding to an edge in G, and has the edge set $\{(u, v) \mid \text{edges in } G \text{ corresponding to vertices } u \text{ and } v \text{ are adjacent}\}$. As an example, a graph G and the line graph $L(G)$ are shown in Fig. 4.1. The following result is due to Dirac [44].

Theorem 4.1 *Every $k \geq 2$ vertices of a k-vertex-connected graph G lie on a circuit of G.*

We now prove the following. Here $P_{x,y}$ refers to the path between vertices x and y.



(a) G (b) $L(G)$

Figure 4.1: (a) Graph G; (b) Line graph $L(G)$

Theorem 4.2 *Given any three vertices x, y and z in a 3-edge-connected graph G, then there exist edge-disjoint paths $P_{x,y}, P_{y,z}$ and $P_{z,x}$ in G.*

Proof: Let $G = (V, E)$ be a 3-edge-connected graph, with x, y and z in V. Form $G^{'}$ by adding three vertices $x^{'}, y^{'}$ and $z^{'}$, and three copies of each edge $xx^{'}, yy^{'}$ and $zz^{'}$. By the edge analogue of the *Expansion Lemma* (adding a new vertex with three edges to old vertices), $G^{'}$ is 3-edge-connected. The line graph $L(G^{'})$ [44] is 3-vertex-connected. By Dirac's Theorem, $L(G^{'})$ has a shortest cycle C through vertices representing $xx^{'}, yy^{'}$ and $zz^{'}$. Since the copies of each added edge have the same closed neighborhood in $L(G')$, this shortest cycle has only one copy each of $xx^{'}, yy^{'}$ and $zz^{'}$. The internal vertices on the three paths joining the vertices $xx^{'}, yy^{'}$ and $zz^{'}$ on C correspond to the desired three paths in G. □

Consider next the graph $G_{n,2}$ shown in Fig. 4.2. This graph has n vertices

$v_1, v_2, \ldots, v_n$. It has the following edges:

$$(v_i, v_j), i = 1, 2, \ldots, n-3 \text{ and } j = i+1, i+2, \text{ and} \tag{4.1}$$

$$(v_{n-2}, v_{n-1}), (v_{n-2}, v_n) \text{ and } (v_{n-1}, v_n). \tag{4.2}$$

Note that the three edges in (4.2) form a complete subgraph on the three vertices v_{n-2}, v_{n-1}, and v_n.

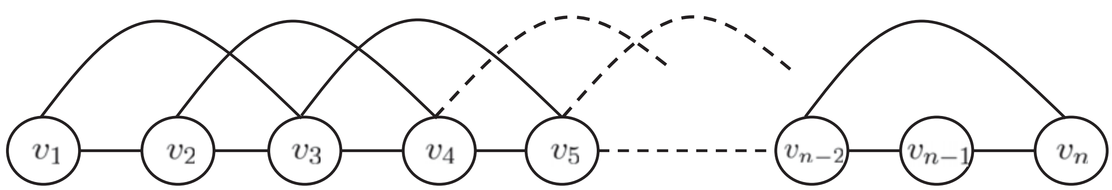


Figure 4.2: Graph $G_{n,2}$

The mapping given in algorithm MAP-$G_{n,2}$ of Fig. 4.3 will be used in the proof of Theorem 4.3.

1: **for** $i = 1, 2, \ldots, n-3$ **do**
2: Map the two edges (v_i, v_{i+1}) and (v_i, v_{i+2}) into mutually disjoint paths in the physical topology.
3: **end for**
4: Map the edges $(v_{n-2}, v_{n-1}), (v_{n-1}, v_n)$ and (v_{n-2}, v_n) into mutually disjoint paths in the physical topology.

Figure 4.3: Algorithm MAP-$G_{n,2}$

We now have the following result.

Theorem 4.3 *The logical graph $G_{n,2}$ in Fig. 4.2 admits a survivable mapping under a single physical edge failure if the physical topology is 3-edge-connected.*

Proof: First we note that the two mutually disjoint paths required in step 2 of MAP-$G_{n,2}$ of Fig. 4.3 exist if the physical topology is 2-edge-connected, and by Theorem 4.2 the three mutually disjoint paths required in step 4 of this mapping exist if the physical topology is 3-edge-connected.

We now show that the mapping MAP-$G_{n,2}$ of Fig. 4.3 is a survivable mapping of $G_{n,2}$, thereby completing the proof of the theorem.

Case 1. Assume that v_{n-1} is not in S and let v_i be the last vertex in the sequence $v_1, v_2, \ldots, v_{n-2}$ that is in S. In this case the vertices v_{i+1} and v_{i+2} will be in $\overline{S}$. So the edges (v_i, v_{i+1}) and (v_i, v_{i+2}) will be in the cut $(S, \overline{S})$.

Case 2. Let v_{n-1} be in S. In this case the edges (v_n, v_{n-1}) and (v_n, v_{n-2}) will be in the cut $(S, \overline{S})$.

Thus, in both cases every cut of $G_{n,2}$ will have two edges that have been mapped by MAP-$G_{n,2}$ into disjoint paths in the physical topology. So, a single physical edge failure will leave at least one edge in every cut, thereby proving (by Theorem 3.5) that the graph $G_{n,2}$ admits a survivable mapping under a single physical edge failure if the physical topology is 3-edge-connected. □

Given a logical topology that does not admit a survivable mapping, we next investigate how this graph can be augmented with new logical links so that the augmented graph is survivable. Our interest is to achieve this without adding protection edges. Note that there are more than one ways to construct a survivable mapping [27]. The procedure for augmentation depends on the algorithm used to construct the survivable mapping. Assuming that the algorithm INCIDENCE-SMART has been used to construct the survivable mapping. Our procedure for augmentation will be as follows.

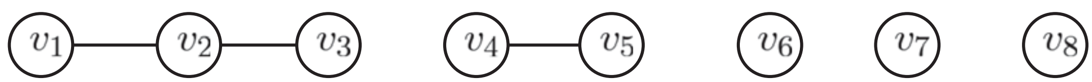
Note that all vertices in the graph G' at the end of the execution of step 2 in algorithm in INCIDENCE-SMART will have degree zero or one. Let V' be the set of

Add additional edges to $G^{'}$ so that it is transformed to $G_{n^{'},2}$ where $n^{'}$ is the number of vertices in $G^{'}$. The original graph G along with the newly added edges is the augmented logical topology.

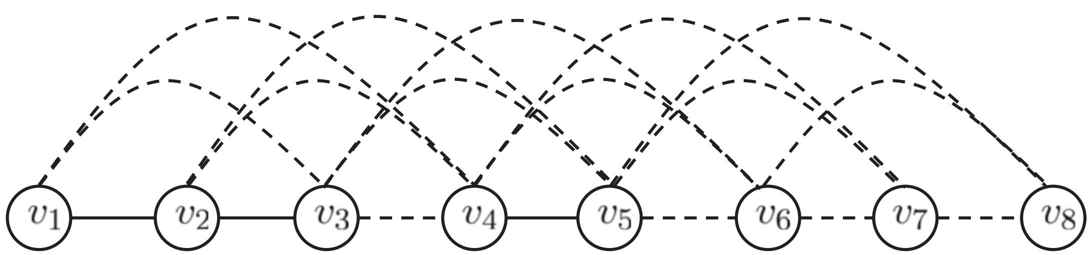
Figure 4.4: Algorithm AUGMENT ($G^{'}$)

vertices in $G^{'}$ and $E^{'}$ be the set of edges in $G^{'}$.

As an example, suppose the graph at the end of step 2 in algorithm INCIDENCE-SMART is as shown in Fig. 4.5a. Then algorithm AUGMENT will produce the augmented graph in Fig. 4.5b.



(a) $G^{'}$



(b) Augmented $G^{'}$

Figure 4.5: (a) Graph $G^{'}$ at the end of INCIDENCE-SMART; (b) Graph after augmentation of $G^{'}$

Given a logical topology G, the following algorithm AUGMENT-MAP-INCIDENCE-SMART uses algorithm INCIDENCE-SMART (Fig. 3.5), algorithm AUGMENT($G^{'}$) (Fig. 4.4) and algorithm MAP-$G_{n,2}$ (Fig. 4.3) to obtain an augmented topology and a mapping of the augmented topology that is survivable under a single physical edge failure, assuming that the physical topology is 3-edge-connected.

Combining the proofs of Theorems 3.6 and 4.3 we obtain the following.

Theorem 4.4 *Given a 2-edge-connected logical topology G_L and a 3-edge-connected physical topology G_P, algorithm AUGMENT-MAP-INCIDENCE-SMART provides an augmentation of G_L and a mapping of the augmented graph that is survivable under a single edge failure in G_P.*

Input: A 2-edge-connected logical topology G and a 3-edge-connected physical topology G_P. Initially, G is the current graph.
Output: An augmented logical topology and a survivable mapping of the augmented topology for single physical edge failure.
1: (Apply step 2 in INCIDENCE-SMART) While there exists a vertex v of degree at least 2 in the current graph, map any two of the edges incident on v into disjoint lightpaths in G_P.
2: Remove v and all the edges incident on v. If there is no vertex of degree 2 in the current graph, go to step 3.
3: Apply algorithm AUGMENT($G^{'}$) on the current graph $G^{'}$. Label the vertices of the augmented graph as in as in Fig. 4.2.
4: Apply algorithm MAP-$G_{n,2}$ on the augmented graph starting first from vertex v_1 in the augmented graph.

Figure 4.6: Algorithm AUGMENT-MAP-INCIDENCE-SMART

4.3 Augmentation for Survivability under Multiple Physical Edge Failures

In this section we generalize the general results of Section 4.2. First, we give a topology and a mapping that needs to be done to guarantee survivability of this topology under multiple physical edge failures. We then show how to augment a given logical topology to achieve survivability under multiple physical edge failures.

The graph $G_{n,k}$ is defined as follows. This graph has n vertices $v_1, v_2, \ldots, v_n$. It has the following edges:

- $(v_i, v_j), i = 1, 2, \ldots, n-k-1$ and $j = i+1, i+2, \ldots, i+k$ and
- the induced subgraph on the $k+1$ vertices $v_{n-k}, v_{n-k+1}, \ldots, v_n$ is a complete graph.

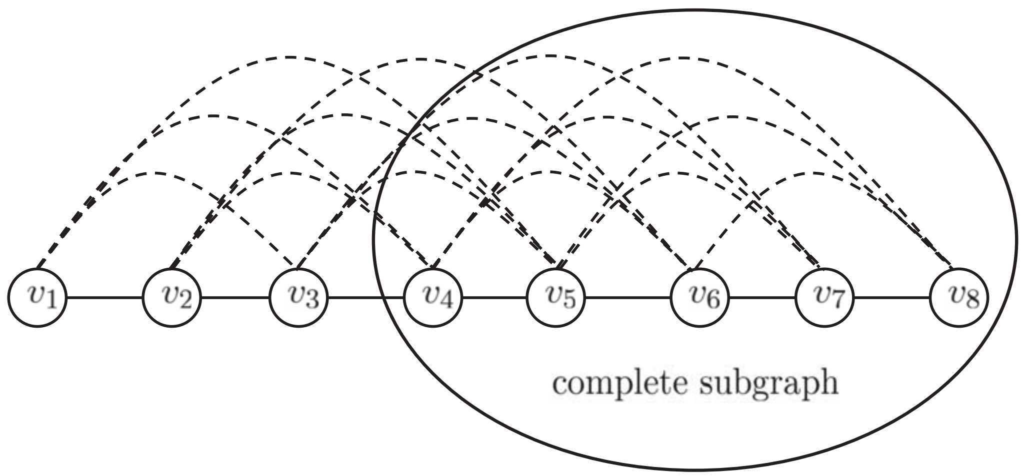


Figure 4.7: Graph $G_{8,4}$

for $i = 1, 2, \ldots, n-k-1$ **do**
    Map the k edges $(v_i, v_j), j = i+1, i+2, \ldots, i+k$ into mutually disjoint paths in the physical topology. (Note: if the physical topology is k-edge-connected, then these k mutually disjoint paths exist.)
    Map the edges in the induced subgraph on the $k+1$ vertices $v_{n-k}, v_{n-k+1}, \ldots, v_n$ into mutually disjoint paths in the physical topology.
end for

Figure 4.8: Algorithm MAP-$G_{n,k}$

As an example, the graph $G_{8,4}$ is shown in Fig. 4.7. And we define algorithm MAP-$G_{n,k}$ in Fig. 4.8, which is a generalization of MAP-$G_{n,2}$.

We now have the following result.

Theorem 4.5 *The logical graph $G_{n,k}$ admits a survivable mapping under $k-1$ physical edge failures if the physical topology is k-edge-connected and there exist mutually disjoint paths in the physical topology connecting the vertices of the logical edges in the complete subgraph induced on the $k+1$ vertices $v_{n-k}, v_{n-k+1}, \ldots, v_n$.*

Proof: We first note that every cut of a complete graph on $k+1$ vertices has at least k edges. And we prove the result by showing that every cut $(S, \overline{S})$ of $G_{n,k}$ has at least k edges that are mapped by algorithm MAP-$G_{n,k}$ into mutually disjoint paths in the physical topology. Then, MAP-$G_{n,k}$ would be a survivable mapping of G_{n-k} tolerating $k-1$ physical edge failures.

Consider a cut $(S, \overline{S})$ of $G_{n,k}$. Assume node n is not in S.

Case 1. All the vertices $v_{n-k}, v_{n-k+1}, \ldots, v_n$ are in $\overline{S}$. In this case, let v_i be the last vertex from the set $\{v_1, v_2, \ldots, v_{n-k-1}\}$ that is in S. Then, by definition, v_i is adjacent to the k vertices $v_{i+1}, v_{i+2}, \ldots, v_{i+k}$ (Fig. 4.7). The corresponding k edges incident on v_i are mapped by algorithm MAP-$G_{n,k}$ into mutually disjoint paths in the physical topology (step 2 in algorithm MAP-$G_{n,k}$). Since the physical topology is k-edge-connected, such mutually disjoint paths exist in the physical topology [44].

Case 2. Let T be the subset of vertices of the set $\{v_{n-k}, v_{n-k+1}, \ldots, v_{n-1}\}$ that are in S. Let $\overline{T}$ be the complement of T in $\{v_{n-k}, v_{n-k+1}, \ldots, v_{n-1}, v_n\}$. Note that these vertices are in $\overline{S}$. Then the set of edges connecting the vertices in T to those in $\overline{T}$ forms a subset of the cut $(S, \overline{S})$. This subset is in fact a cut of the complete subgraph on the $k+1$ vertices $v_{n-k}, v_{n-k+1}, \ldots, v_n$ and has at least k edges.

Thus, we have proved that every cut $(S, \overline{S})$ of $G_{n,k}$ has at least k edges that are mapped by algorithm MAP-$G_{n,k}$ into mutually disjoint paths in the physical topology. This guarantees that every cut of $G_{n,k}$ will have at least one edge after $k-1$ physical edge failures, thereby demonstrating that MAP-$G_{n,k}$ provides a survivable mapping of $G_{n,k}$ under $k-1$ physical edge failures, if the conditions of the theorem are satisfied. □

We next give a generalized version of algorithm AUGMENT-MAP-INCIDENCE-SMART that achieves an augmentation of a logical topology and provides a survivable mapping of the augmented logical topology under $k-1$ physical edge failures, provided certain requirements are satisfied. In this algorithm the augmentation procedure given in Fig. 4.9 is used.

1: Add additional edges to G so that it is transformed to $G_{n,k}$ where n is the number of vertices in G. The original graph along with the newly added edges is the augmented logical topology.

Figure 4.9: GENERAL-AUGMENT (G)

Input: A k-edge-connected logical topology G and a k-edge-connected physical topology G_P. Initially G is the current graph.
Output: An augmented logical topology and a survivable mapping of the augmented topology that tolerates $k-1$ physical edge failures.
1: While there exists a vertex v of degree at least k in the current graph, map any k of the edges incident on v into disjoint lightpaths in G_P. Remove v and all the edges incident on v. If there exists no vertex of degree k in the current graph, go to step 2.
2: Apply algorithm GENERAL-AUGMENT (G') on the current graph G'.
3: Label the vertices of the augmented topology (as in Fig. 4.7). Apply algorithm MAP-$G_{n,k}$ on the augmented graph, starting from vertex v_1.

Figure 4.10: GENERAL-AUGMENT-MAP-INCIDENCE-SMART

Combining the proofs in Theorems 3.6 and 4.5 we obtain the following.

Theorem 4.6 *Given a k-edge-connected logical topology G_L and a k-edge-connected physical topology G_P, algorithm GENERAL-AUGMENT-MAP-INCIDENCE-SMART (Fig. 4.10) provides an augmentation of G_L and a mapping of the augmented graph that is survivable under $k-1$ edge failures in G_P, provided there exist mutually disjoint paths connecting the vertices of the logical edges of the complete subgraph induced on the $k+1$ vertices $v_{n-k}, v_{n-k+1}, \ldots, v_n$.*

Algorithm MAP-$G_{n,k}$ (Fig. 4.8) requires that we be able to find mutually disjoint paths in the physical topology for the $k(k+1)/2$ edges of the complete subgraph of $G_{n,k}$ on the vertices $v_{n-k}, v_{n-k+1}, \ldots, v_n$. One cannot guarantee the existence of such paths. Suppose we replace step 3 in algorithm MAP-$G_{n,k}$ by step $3'$ as follows.

Step $3'$: For $i = n-k, n-k+1, \ldots, n-1,$ map the edges (v_i, v_j),

$j = i+1, i+2, \ldots, n$ into mutually disjoint paths in the physical topology.

Nodes/Failures	2	3	4	5	6
10 nodes	0.884	0.9482	0.983	0.9876	0.9938
20 nodes	0.9254	0.9604	0.9908	0.9946	0.9986
30 nodes	0.9334	0.9584	0.9932	0.9954	0.9984
40 nodes	0.9402	0.9638	0.9944	0.9946	0.999
50 nodes	0.9364	0.9732	0.9944	0.9956	0.9998
60 nodes	0.9446	0.9724	0.9952	0.9956	0.999
70 nodes	0.938	0.9768	0.996	0.9974	0.9998
80 nodes	0.937	0.9782	0.996	0.9968	0.9992
90 nodes	0.9428	0.9788	0.9948	0.9966	0.9996
100 nodes	0.9414	0.9784	0.9968	0.9982	0.999

Table 4.1: Survivability index

Then, let us investigate how well this modified mapping is able to help tolerate $k-1$ physical edge failures. Towards this end, we define the *survivability index* of a mapping $\prod$ with respect to a logical topology G_L as the fraction of failure patterns of a specified size under which the given logical topology remains connected when the logical links are mapped by $\prod$. We performed extensive simulations on $G_{n,k}$ for different values of n and k. In each case the physical topology is chosen as a k-connected graph generated by the procedure given in [41] (Chapter 8). We considered 5000 randomly generated physical link failure patterns of size $k-1$ and checked if the logical topology remains connected after the occurrence of each failure pattern. Using the number of times the logical topology tolerated the failure patterns, the survivability index is calculated in each case and is given in the table of Table 4.1.

We note that for a fixed value of k, the survivability index of the modified mapping of $G_{n,k}$ increases with n. This is because, as n increase, the number of cuts of $G_{n,k}$ that are affected by the edges involved in the modified step $3'$ decreases, thereby increasing the survivability index. For a fixed value of n, the survivability index also increases as the value of k increases. This is because, as the value of k increases, the connectivity (and hence density) of the physical topology also increases, thereby decreasing the probability of picking a failure pattern under which the logical topology gets disconnected. Overall, we find that the survivability index of the modified mapping is quite high. In other words, relaxing the requirement of step 3 in MAP-$G_{n,k}$ does not result in a significant reduction in the survivability index.

4.4 Summary

Given a logical topology in an IP-over-WDM optical network, we investigated the problem of augmenting this topology with additional links so that the augmented topology admits a mapping under which it remains connected when one or more physical link failures occur. We identified a special logical topology structure that can be used to achieve the required augmentation. The structure of this topology depends on the number of physical link failures that are required to be tolerated. An interesting future direction of research is to identify other structures that can be used to achieve the augmentation. In doing so, we also need to make sure that the number of additional links to be added is as small as possible.

The work in this chapter has been reported in [32][37].

Chapter 5

Robustness of Logical Topology Mapping Algorithms for Survivability against Multiple Failures in an IP-over-WDM Optical Network

5.1 Introduction

The SMART-based algorithms – CIRCUIT-SMART, CUTSET-SMART, CUTSET-SMART-SIMPLIFIED, and INCIDENCE-SMART reported in [27] and discussed in Chapter 3 – are all designed to provide survivability at the logical layer against a single physical edge failure. Also, a drawback of CUTSET-SMART and INCIDENCE-SMART algorithms is that, they require the augmentation of the graph with additional links to guarantee single-layer survivability. The augmentation problem was considered in Chapter 4. In this chapter we first draw attention to a short-coming of the CUTSET-SMART algorithm described in Chapter 3. We then present GEN-CUTSET-SMART described in [28] to overcome this short-coming. We follow this by an introduction of the concept of robustness of an algorithm which captures the ability of the algorithm to provide survivability against multiple physical failures. This is similar to the concept of fault coverage used in hardware/software testing. We analyze the different algorithmic frameworks for their robustness property. Using simulations, we demonstrate that even when an algorithm cannot be guaranteed to provide survivability against multiple failures, its robustness could be very high. The work also provides a basis for the design of survivability mapping algorithms when special classes of failures such as SRLG failures are to be protected against.

5.2 GEN-SMART: A Generalized Algorithmic Framework for the SLTM Problem

We first draw attention to a shortcoming of the algorithmic framework CUTSET-SMART. Algorithm CIRCUIT-SMART of [27] would not require additional edges to be added to the logical graph if no new edges (protection edges) are added in step 3 of this algorithm. This is not the case with algorithm CUTSET-SMART. This algorithm requires protection edges to be added to all unmapped branches. So, CIRCUIT-SMART guarantees a survivable mapping of the given logical graph, if step 3 does not require any new edges to be added. On the other hand, CUTSET-SMART guarantees a survivable mapping of the graph obtained by contracting the unmapped branches in the logical graph, if step 3 of this algorithm does not require any new edges to be added.

The question now arises if it is possible to obtain a generalized version of CUTSET-SMART that does not have this limitation. The rest of the section addresses this question and provides an affirmative answer. This section is based on [28].

An ordered sequence $Q(b_1), Q(b_2), \ldots, Q(b_k)$ is a *generalized cutset cover sequence*

- if this sequence is a cutset cover sequence, and
- for every unmapped branch b_i, $Q(b_i) \bigcap \hat{S}(b_j) = \hat{S}(b_j)$, where j is the largest index such that $Q(b_i) \bigcap \hat{S}(b_j) \neq \emptyset$. In this case we say that the unmapped branch b_i is covered by the branch b_j. We also say that branch b_j covers itself.

Given a generalized cutset cover sequence $Q(b_1), Q(b_2), \ldots, Q(b_k)$, we define the set $Q - Cover(b_i)$ for each $i = 1, 2, \ldots, k$ as the set of all branches (including itself) covered by the branch b_i. The $Q - Cover$ sets define a partition of the branches of the given spanning tree. If we arrange the rows of the f-cutset matrix to correspond to the sets $Q - Cover(b_1), Q - Cover(b_2), \ldots, Q - Cover(b_k)$ in that order and arrange the columns to correspond to the sets $Q - Cover(b_1), Q - Cover(b_2), \ldots, Q - Cover(b_k), \hat{S}(b_1), \hat{S}(b_2), \ldots, \hat{S}(b_k)$, then the f-cutset matrix will have the form shown in (5.1). In this figure, I stands for a matrix of all 1's, O is a matrix of 0's and U refers to the unit matrix of appropriate size. Also $Q_c(b_i)$ stands for $Q - Cover(b_i)$. In [28] an algorithm is given to construct a generalized cutset cover sequence starting from a cutset cover sequence.

$$
\begin{array}{cccccccccccccc}
Q_c(b_1) & Q_c(b_2) & Q_c(b_3) & \cdots & Q_c(b_{k-1}) & Q_c(b_k) & \hat{S}(b_1) & \hat{S}(b_2) & \hat{S}(b_3) & \cdots & \cdots & \cdots & \hat{S}(b_{k-1}) & \hat{S}(b_k) \\
\hline
U & O & O & \cdots & O & O & I & O & O & \cdots & O & \cdots & O & O \\
O & U & O & \cdots & O & O & \times & I & O & \cdots & O & \cdots & O & O \\
O & O & U & \cdots & O & O & \times & \times & I & \cdots & O & \cdots & O & O \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
O & O & O & \cdots & U & O & \times & \times & \times & \times & \times & \times & I & O \\
O & O & O & \cdots & O & U & \times & \times & \times & \times & \times & \times & O & I
\end{array}
\tag{5.1}
$$

Given a spanning tree of a logical graph, we now present a generalized version of CUTSET-SMART called GEN-SMART (Fig. 5.1) that does not require addition of protection edges to the unmapped branches. This algorithm does not have step 5 of CUTSET-SMART and has a modified version of step 2 and 3 of CUTSET-SMART.

This framework shown in Fig. 5.1 includes as special cases the other SMART-based algorithms discussed in [28]. For the sake of simplicity in presentation we have assumed in the description of GEN-SMART that all the edges in the set $A \subseteq \hat{S}(b_i)$ and $B \subseteq Q - Cover(b_i)$ can be mapped into disjoint paths in G_P. But this may not always be possible. In such cases, we map a maximum subset of these edges into disjoint paths. To the other edges in this set we add protection edges and map each edge and its protection edge into disjoint paths in G_P. Also, if we choose $A = \hat{S}(b_i)$ and $B = Q - Cover(b_i)$ then GEN-SMART becomes the same as GEN-CUTSET-SMART presented in [28]. Also, different choices of A and B in GEN-SMART lead to different versions of SMART-based algorithms discussed in earlier works. These choices and the corresponding versions are given next.

```
Starting with any cutset cover sequence generate a generalized cutset cover sequence of G_L. Let this sequence be Q(b_1), Q(b_2), ..., Q(b_k).
for i = 1, 2, ..., k do
    Let A ⊆ Ŝ(b_i) and B ⊆ Q − Cover(b_i)
    Map the edges in the set b_i ∪ A ∪ B into disjoint lightpaths in G_P.
end for
```

Figure 5.1: Algorithm GEN-SMART

Choice of A and B	Special case of GEN-SMART
$\lvert A\rvert = 1, \lvert B\rvert = 1$	CUTSET-SMART-SIMPLIFIED
$A = \hat{S}(b_i)$, $\lvert B\rvert = 1$	CUTSET-SMART
$\lvert A\rvert = 1, B = Q - Cover(b_i)$	CIRCUIT-SMART
$A = \hat{S}(b_i), B = Q - Cover(b_i)$	GEN-CUTSET-SMART

Table 5.1: Special cases of GEN-SMART algorithms

```
Starting with any cutset cover sequence generate a generalized cutset cover sequence of G_L. Let this sequence be Q(b_1), Q(b_2), ..., Q(b_k).
for i = 1, 2, ..., k do
    Pick a chord c in Ŝ(b_i).
    Map the edges b_i and c into disjoint lightpaths in G_P.
end for
```

Figure 5.2: Algorithm CUTSET-SMART-SIMPLIFIED

```
Starting with any cutset cover sequence generate a generalized cutset cover sequence of G_L. Let this sequence be Q(b_1), Q(b_2), ..., Q(b_k).
for i = 1, 2, ..., k do
    Map the edges in the set b_i ∪ Ŝ(b_i) into disjoint lightpaths in G_P.
end for
```

Figure 5.3: Algorithm CUTSET-SMART

For the sake of completeness, we repeat these special versions in Figs. 5.2–5.4. See also Table 5.1. Some important observations on the different versions of GEN-SMART are now in order:

1: Starting with any cutset cover sequence generate a generalized cutset cover sequence of G_L. Let this sequence be $Q(b_1), Q(b_2), \ldots, Q(b_k)$.
2: **for** $i = 1, 2, \ldots, k$ **do**
3: Pick a chord c in $\hat{S}(b_i)$.
4: Map the edges in the set $c \bigcup Q - Cover(b_i)$ into disjoint lightpaths in G_P.
5: **end for**

Figure 5.4: Algorithm CIRCUIT-SMART

1: Starting with any cutset cover sequence generate a generalized cutset cover sequence of G_L. Let this sequence be $Q(b_1), Q(b_2), \ldots, Q(b_k)$.
2: **for** $i = 1, 2, \ldots, k$ **do**
3: Map the edges in the set $\hat{S}(b_i) \bigcup Q - Cover(b_i)$ into disjoint lightpaths in G_P.
4: **end for**

Figure 5.5: Algorithm GEN-CUTSET-SMART

- CUTSET-SMART-SIMPLIFIED, the simplest of all these algorithms, does not guarantee survivability even against a single physical link failure, unless protection edges are added to the unmapped branches [27][28].

- CUTSET-SMART does not guarantee survivability even against a single physical link failure, unless protection edges are added to the unmapped branches [28]. But it has potential to provide some degree of survivability against multiple failures.

- CIRCUIT-SMART guarantees survivability against a single failure [26][27], but its potential to provide survivability against multiple failures is limited.

- GEN-CUTSET-SMART guarantees survivability against a single failure, and its potential to guarantee survivability against multiple failures is very high.

Both CUTSET-SMART and GEN-CUTSET-SMART have higher potential to provide survivability against multiple failures because in both these algorithms all the edges in $\hat{S}(b_i)$ are mapped. In the next section we provide an analytical evaluation of the extent to which these algorithms provide survivability against multiple failures.

5.3 Robustness of Survivable Logical Topology Mapping Algorithms

In this section we first define the concept of robustness of an algorithm that is a measure of the ability of the algorithms to provide survivability against multiple physical failures.

Given a logical topology G_L and a physical topology G_P, the robustness $\beta(A, r)$ of a logical topology mapping algorithm A with respect to G_P and G_L is defined as the ratio of the number of cuts of G_L that are protected by algorithm A against r physical link failures to the total number of cuts in G_L.

For these algorithms we now proceed to evaluate $\beta(A, r)$. In the following A_1, A_2, A_3 and A_4 denote algorithms CUTSET-SMART-SIMPLIFIED, CUTSET-SMART, CIRCUIT-SMART, and GEN-CUTSET-SMART, respectively.

Given a generalized cutset cover sequence $Q(b_1), Q(b_2), \ldots, Q(b_k)$. Let us first partition all cuts in G_L into the sets $Q_1, Q_2, \ldots, Q_k$ where Q_i is the set of all cuts that contain at least one branch from the set $Q - Cover(b_i)$ and no branch from any set $Q - Cover(b_j)$, $j > i$. Note that this partition is well defined since every cut must have at least one branch.

Consider now a cut $S \in Q_i$. Assume that S contains p branches from Q_i. Now we recall the following results from [27][28].

Theorem 5.1 *Given a cutset cover sequence* $Q(b_1), Q(b_2), \ldots, Q(b_k)$, *let* $Q(b_{i_1}), Q(b_{i_2}), \ldots, Q(b_{i_l})$ *be a subsequence of this sequence then* $\hat{S}(b_{i_l}) \subseteq Q(b_{i_1}) \bigoplus Q(b_{i_2}) \bigoplus \ldots \bigoplus Q(b_{i_l})$.

In view of Theorem 5.1, the cut S will have the form in Fig. 5.6 if S has an odd number p of branches from the set $Q - Cover(b_i)$. Note that if p is even then none of the chords in $\hat{S}(b_i)$ will be in S. The numbers of edges mapped disjointly by the

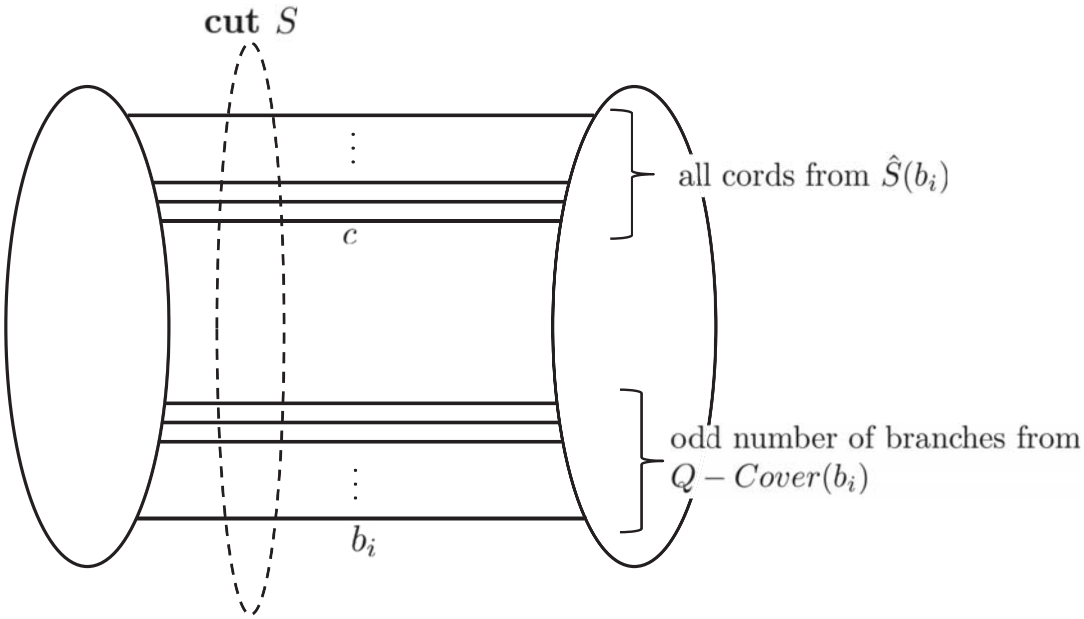


Figure 5.6: Cut S

different Algorithms A_1, A_2, A_3, and A_4 are:

- Algorithm A_1 maps b_i and a chord c in $\hat{S}(b_i)$ disjointly, if S contains b_i.
- Algorithm A_2 maps b_i and all edges in $\hat{S}(b_i)$ disjointly, if S contains b_i.
- Algorithm A_3 maps all the p branches and a chord c in $\hat{S}(b_i)$.
- Algorithm A_4 maps all the p branches and all the chords in $\hat{S}(b_i)$.

Thus we have the following:

- Algorithm A_1 protects S against at least one physical link failure, if S contains b_i.
- Algorithm A_2 protects S against at least $|\hat{S}(b_i)|$ physical link failures, if S contains b_i.
- Algorithm A_3 protects S against at least p physical link failures.

- Algorithm A_4 protects S against at least $p + |\hat{S}(b_i)| - 1$ physical link failures.

Since $p \geq 1$, we can restate the last statement as:

- Algorithm A_4 protects S against at least $|\hat{S}(b_i)|$ physical link failures.

Let us now calculate the total number of cuts in Q_i that has an odd number of branches from the set $Q - Cover(b_i)$. Let this number be denoted as $ODD(Q_i)$.

Let $h_i = |Q - Cover(b_i)|, g_i = |\hat{S}(b_i)|$. $h = \min h_i$ and $g = \min g_i$. Also, let $N_i = h_1 + h_2 + \ldots + h_i$.

Robustness of Algorithm A_1:

Algorithm A_1 will protect against a single physical failure all cuts from each Q_i that have an odd number of branches from the set $Q - Cover(b_i)$ and contain branch b_i. This number is

$$\begin{aligned}
&= (\text{Number of combinations of branches from the sets } Q - Cover(b_k), \\
&\quad k = 1, 2, \ldots, i-1) \times (\text{Number of combinations of odd number of} \\
&\quad \text{branches from the set } Q - Cover(b_i) \text{ that contain } b_i) \\
&= 2^{N_{i-1}} \times 2^{h_i - 2} \\
&= 2^{N_i}/4.
\end{aligned}$$

Since the number of cuts in G_L is $2^{n-1} - 1$, where n is the number of nodes in G_L, and $n - 1 = h_1 + h_2 + \ldots + h_k$, we get

$$\beta(A_1, 1) \geq 1/4(\sum_{i=1}^{k} 2^{N_i})/(2^{n-1} - 1). \tag{5.2}$$

Note that if $p \geq 2, \beta(A, p) \geq 0$, since there is no guarantee that algorithm A_1 will protect any cut if 2 or more physical failures occur.

Robustness of Algorithm A_2:

Algorithm A_2 will protect against g_i physical failures all cuts from each Q_i that have an odd number of branches from the set $Q - Cover(b_i)$ and contain branch b_i. This follows from the fact that each such cut will have b_i and all edges in $\hat{S}(b_i)$ that are mapped disjointly. So,

$$\beta(A_2, g) \geq 1/4(\sum_{i=1}^{k} 2^{N_i})/(2^{n-1} - 1). \tag{5.3}$$

Robustness of Algorithm A_3:

Algorithm A_3 will protect against at least p physical failures all cuts from each Q_i that have an odd number p of branches from the set $Q - Cover(b_i)$. This follows from the fact that each such cut will have p branches and at least one chord c in $\hat{S}(b_i)$ that are mapped disjointly. This number is

= (Number of combinations of branches from the sets $Q - Cover(b_k)$,

$k = 1, 2, \ldots, i-1$) $\times$ (Number of combinations of branches from the set $Q - Cover(b_i)$)

$= 2^{N_{i-1}} C(h_i, p)$.

So

$$\beta(A_3, p) \geq \left(\sum_{i=1}^{k} 2^{N_i - 1} \sum_{\text{odd } q \geq p}^{h_i} C(h_i, q) \right) / (2^{n-1} - 1) \quad \text{for odd } p \geq 1 \tag{5.4}$$

where $C(h_i, q)$ is the number of q-combinations of h_i elements.

If $p = 1$, then it can be verified that $\beta(A_3, 1) = 1$, confirming that CIRCUIT-SMART protects G_L against any single physical link failure [26][27].

Robustness of Algorithm A_4:

Algorithm A_4 will protect against at least $\left|\hat{S}(b_i)\right|$ physical failures all cuts from each Q_i that have an odd number of branches from the set $Q-Cover(b_i)$. This follows from the fact that each such cut will have at least one branch and all the chords in $\hat{S}(b_i)$ that are mapped disjointly. This number is

$$\begin{aligned}
&= (\text{Number of combinations of branches from the sets } Q - Cover(b_k), \\
&\quad k = 1, 2, \ldots, i-1) \times (\text{Number of combinations of } p \text{ branches from the} \\
&\quad \text{set } Q - Cover(b_i)) \\
&= 2^{N_{i-1}} \times 2^{h_i - 1} \\
&= 2^{N_i}/2.
\end{aligned}$$

So

$$\beta(A, p) \geq 1/2(\sum_{i=1}^{k} 2^{N_i})/(2^{n-1} - 1). \tag{5.5}$$

Let $SUM = \left(\sum_{i=1}^{k} 2^{N_i}\right)/(2^{n-1} - 1)$. Then we can rewrite (5.2), (5.3), (5.5) as

- $\beta(A_1, 1) \geq 1/4\ SUM.$
- $\beta(A_2, g) \geq 1/4\ SUM.$
- $\beta(A_4, g) \geq 1/2\ SUM.$

The value of SUM depends on the choice of generalized cutset cover sequence. The lower bounds in the above are the numbers of cuts that are guaranteed to be protected by the respective algorithms. Depending on the length of the generalized cutset cover sequence, the sizes of h_i's and g_i's, the location of physical link failures and the mappings used, the number of protected cuts could be much larger. The higher the value of $\beta(A, r)$ the higher will be the probability that algorithm A will protect G_L from any set of r physical link failures.

5.4 Simulation Results and Analysis

To compare the performance of CUTSET-SMART-SIMPLIFIED, CIRCUIT-SMART, CUTSET-SMART, and GEN-CUTSET-SMART with respect to their ability to provide multiple failure survivability simulation studies were conducted using LEMON (Library for Efficient Modeling and Optimization in Networks) [45] and G++ under Linux system. The physical and logical topologies were regular topologies with connectivity equal to 3, 4, and 5 constructed using a procedure originally given by Harary and described in [41]. The number of nodes in the physical topologies was set to 50, 60, 70, 80, 90, and 100 nodes. The nodes in logical topologies were a subset of the physical nodes and the number of nodes in a logical topology was set to 50% of the nodes in the corresponding physical topology.

For each combination of (topology connectivity, number of nodes in physical topology, number of physical link failures), 100 physical and corresponding logical topology pairs were generated and tested against 4 algorithms described in the previous section. Given k-connected physical and logical topologies, the survivability of the G_L under multiple (2 to $k-1$) physical link failures is determined by the number of G_L's which remain connected against physical link failures. Our simulation enumerated all possible combinations of physical link failures and evaluated how many G_L's could remain connected. The success rate in each case is calculated.

First a spanning tree on a logical topology was generated and the fundamental circuits and cutsets with respect to the spanning tree were found. The generalized cutset cover sequence was generated using the algorithms in [28]. With the information of the fundamental cutsets, the $Q - Cover(b_i)$ and $\hat{S}(b_i)$ sets were generated. Then we applied the four algorithms (CUTSET-SMART-SIMPLIFIED, CIRCUIT-SMART, CUTSET-SMART, and GEN-CUTSET-SMART) and mapped maximal number of edges disjointly in $b_i \bigcup A \bigcup B$. If the disjoint mappings for some of the edges in

3-conn	50 nodes		60 nodes		70 nodes	
failures \ Algorithms	1	2	1	2	1	2
A_1	92.173	71.857	89.711	65.294	89.429	64.338
A_2	92.987	73.701	90.533	67.080	90.371	66.024
A_3	100	85.367	100	83.775	100	82.263
A_4	100	86.426	100	84.406	100	83.375
3-conn	80 nodes		90 nodes		100 nodes	
failures \ Algorithms	1	2	1	2	1	2
A_1	87.617	57.744	86.570	55.356	84.427	52.313
A_2	88.700	59.710	87.963	57.2	85.853	54.405
A_3	100	78.811	100	78.377	100	76.149
A_4	100	79.913	100	79.367	100	77.073

Table 5.2: Success rate for 3-connected physical and logical topologies

$b_i \bigcup A \bigcup B$ do not exist, a parallel edge is added to the logical topology and the newly added edge is mapped disjointly with the original edge. At the end of the procedure, the unmapped logical edges were randomly mapped, which could increase the chance of survivability for the logical mapping.

The simulation results giving the success rate are shown in Table 5.2, 5.3, and 5.4. Notice that in Table 5.2, extra tests for the single failure case in 3-connected physical and logical topologies are presented, which show that CUTSET-SMART and GEN-CUTSET-SMART can guarantee 100% survivability for the logical topology under a single physical link failure, while CUTSET-SMART-SIMPLIFIED and CIRCUIT-SMART can not.

Based on the simulations, we summarize our observations as follows.

- The value of SUM is at most 2. This can be reached when each $h_i = 1$. In such cases, (5.2) and (5.3) simplify to $\beta(A_1, 1) \geq 1/2, \beta(A_2, g) \geq 1/2$. In spite of this low value on the corresponding robustness, algorithms A_1 and A_2 have higher ability to provide survivability against multiple physical link failures.

- As expected, A_2 has higher potential to provide survivability against multiple

4-conn	50 nodes		60 nodes		70 nodes	
failures \ Algorithms	2	3	2	3	2	3
A_1	94.709	85.841	93.907	84.533	93.655	81.356
A_2	95.975	88.679	95.272	86.979	94.841	84.513
A_3	96.646	88.262	95.950	87.549	95.383	85.219
A_4	97.367	90.263	96.665	89.159	96.235	86.984
4-conn	80 nodes		90 nodes		100 nodes	
failures \ Algorithms	2	3	2	3	2	3
A_1	92.381	80.498	91.575	78.445	91.000	76.815
A_2	94.018	83.343	93.373	81.564	93.043	79.780
A_3	94.801	83.473	93.983	81.802	93.466	79.700
A_4	95.639	85.396	95.018	83.819	94.41	81.582

Table 5.3: Success rate for 4-connected physical and logical topologies

5-conn	50 nodes		60 nodes		70 nodes	
failures \ Algorithms	2	3	2	3	2	3
A_1	99.764	99.450	99.785	99.366	99.809	99.246
A_2	99.912	99.653	99.880	99.634	99.888	99.583
A_3	99.877	99.617	99.869	99.541	99.867	99.473
A_4	99.956	99.810	99.935	99.771	99.937	99.746
5-conn	80 nodes		90 nodes		100 nodes	
failures \ Algorithms	2	3	2	3	2	3
A_1	99.772	99.231	99.668	99.184	99.674	99.089
A_2	99.858	99.557	99.785	99.510	99.787	99.507
A_3	99.848	99.827	99.827	99.437	99.804	99.363
A_4	99.916	99.915	99.915	99.725	99.899	99.654

Table 5.4: Success rate for 5-connected physical and logical topologies

failures compared to A_1.

- As expected, algorithms A_3 and A_4 have higher success rate compared to A_1 and A_2.

- The success rate of all algorithms is higher for higher values of connectivity of physical topologies. This could be due to the survivability of a large number of disjoint paths. This calls for future research.

5.5 Conclusion

The SLTM problem in an IP-over-WDM optical network is to map each link (u, v) in the logical topology G_L into a lightpath between the nodes u and v in the physical topology G_P such that failure of a physical link does not cause the logical topology to become disconnected. It is assumed that both the physical and logical topologies are 2-edge-connected. Most research in this area has focused on logical topology survivability against a single physical link failure. Also, existing approaches do not provide insight into the problem when multiple physical link failures, such as SRLG failures, occur.

In this chapter we pursued the structural approach developed in [26][27][28] to study the logical topology mapping problem for the case of multiple failures. We first presented a generalized algorithmic framework for the SLTM problem. This framework includes several other frameworks considered in earlier works [27][28] as special cases. We then defined the concept of robustness of a mapping algorithm which captures the ability of the algorithm to provide survivability against multiple physical link failures. This is similar to the concept of fault coverage used in hardware/software testing. The higher the value of the robustness of an algorithm the higher the probability that the algorithm will be able to provide survivability.

We analyzed the different frameworks for their robustness property. Specifically, we provided lower bounds for the robustness for the different algorithms. These lower bounds give the number of cuts which an algorithm is guaranteed to protect against multiple failures. The quantity *SUM* used in these formulas depends on several structural features such as the choice of the generalized cutset cover sequence to be used to provide higher degree of robustness.

Using simulations, we demonstrate that even when an algorithm cannot be guaranteed to provide survivability against all multiple failures, its robustness could be very high. The work also provides a basis for the design of survivable mapping algorithms when special classes of failures such as SRLG failures are to be protected against. Further work along these lines is in progress.

The work in this chapter has been reported in [38].

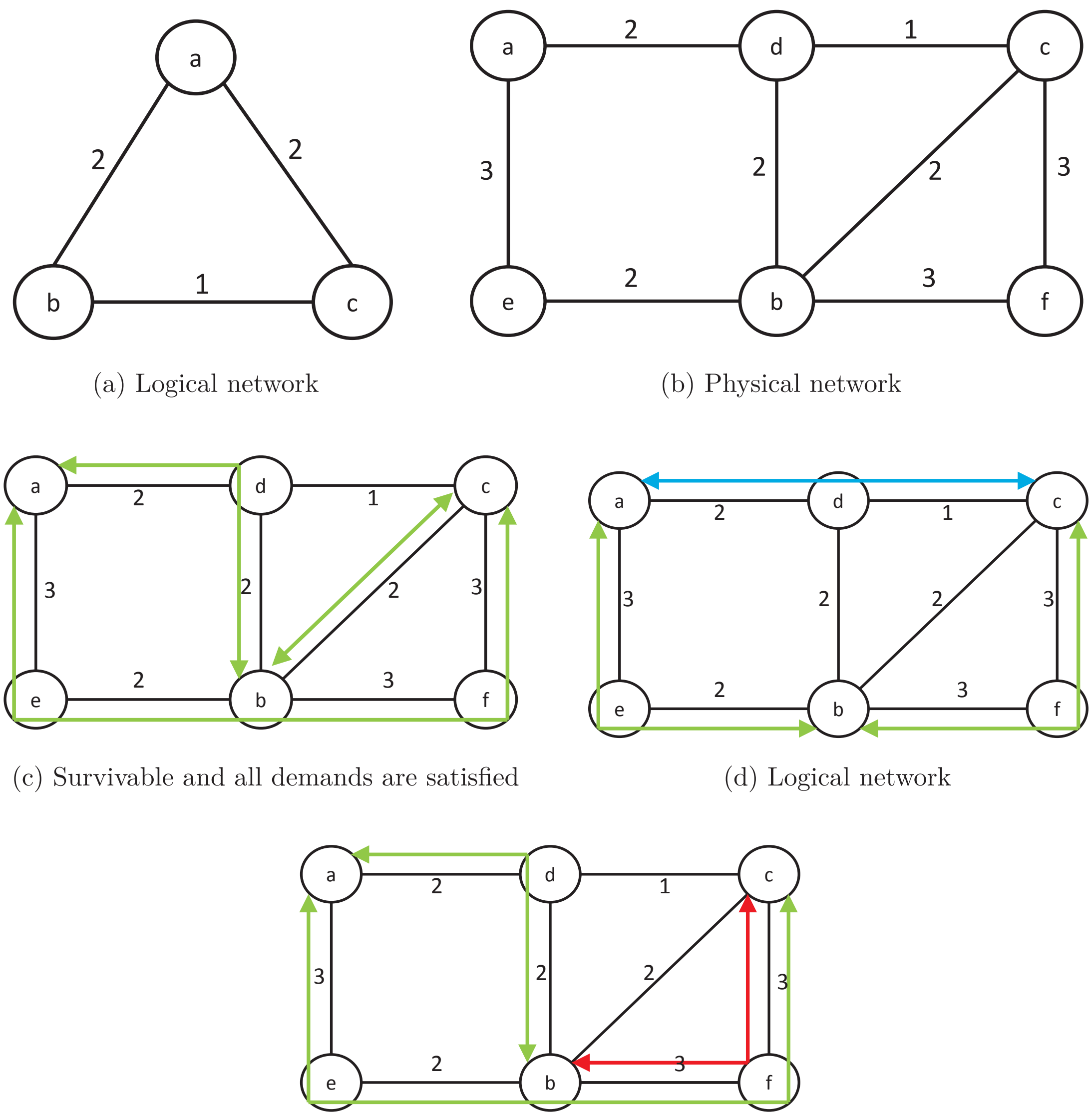
Chapter 6

Logical Topology Survivability in IP-over-WDM Networks: Survivable Lightpath Routing for Maximum Logical Topology Capacity and Minimum Spare Capacity Requirements

6.1 Introduction

Most previous research concentrated on survivable design of un-capacitated IP-over-WDM networks, while in practice, physical link capacities and logical link demands are usually considered during design phase to reduce costs. In this chapter we consider survivable logical topology design in IP-over-WDM networks with capacity and demand constraints on physical and logical links, respectively. For un-capacitated IP-over-WDM networks, survivability is achieved if the logical network remains connected after any physical link failure. In such a case, since the logical network will be connected after a physical link failure, the existence of alternative lightpaths for the failed logical links is guaranteed. However, if the physical link capacity is taken into consideration, demands on logical links may not be satisfied after physical link failure(s) even if the logical network remains connected. Thus, the original definition of survivability in un-capacitated IP-over-WDM networks does not apply to capacitated networks. In order to satisfy demands on logical links we need to add spare capacity to each physical link, which is the extra capacity required to carry the disrupted traffic.

Figures 6.1a and 6.1b show a logical network with demands on its links and a physical network with capacities on its links. A survivable routing satisfying both logical link demands and guaranteeing logical graph survivability after a single physical link failure is shown in Fig. 6.1c. For the mappings in Figs. 6.1d and 6.1e, either the logical topology survivability criterion or the logical demand constraints will not be satisfied after a physical link failure.



(a) Logical network

(b) Physical network

(c) Survivable and all demands are satisfied

(d) Logical network

(e) Physical network

Figure 6.1: Capacitated survivability and demand satisfaction

In this chapter we define a capacitated IP-over-WDM network to be *weakly sur-*

vivable if there exists a mapping such that the logical network remains connected after a single physical link failure. Note that under weak survivability, not all the logical link demands need to be satisfied after a physical link failure. We define a capacitated IP-over-WDM network to be *strongly survivable* if there exists a logical topology mapping that satisfies two criteria: the logical network remains connected after any physical link failure, and there exists sufficient capacity on physical links to support all disrupted traffic. In this chapter, we provide exact MILP formulations and heuristics for the strongly and weakly survivable mappings in capacitated IP-over-WDM networks. We also consider the issue of spare capacity assignment at the physical layer to achieve strong survivability.

The rest of this chapter is organized as follows. Section 6.2 provides a brief review of related literature. Formal definitions of weak and strong survivability and notations are presented in Section 6.3. This section also defines two classes of problems considered in this chapter. Section 6.3.1 and 6.3.2 provide exact solutions for the two scenarios. We develop heuristics, present experimental settings, and provide a comparative evaluation of the MILP approaches and the heuristics in Section 6.4. Section 6.5 discusses how the approach in Section 6.3 can be extended to accommodate different performance criteria.

6.2 Literature Review

Extensions of the work in [5] are given in [46] and [22]. Lee and Modiano [46] introduced certain connectivity metrics for layered networks and provided ILP formulations for the lightpath routing problem satisfying these metrics. In particular, they provided approximation heuristics for lightpath routing maximizing the min cross layer cut metric. This metric captures the robustness of the networks after multiple physical link failures. Kan et al. [22] discussed the relationship between survivable lightpath routing and the spare capacity requirements on the logical links to satisfy the original traffic demands after failures. A common drawback of ILP approaches is that they are not scalable as the network size increases. Hence, heuristic approaches that provide approximations to the optimal solutions are presented.

There has been a great deal of research on the single layer network survivability problem, in particular, assignment of spare capacities on the physical links to guarantee the required network flows after link failures. Some recent works in this area are [47] and [48]. Some of the other works that studied the spare capacity assignment problem under survivability requirements are [49] and [50]. All these works do not consider the notion of survivability of the logical layer that is critical in IP-over-WDM networks. As remarked earlier, Kan et al. [22] discussed the relationship between survivable lightpath routing and spare capacity requirements on the logical links to satisfy the original traffic demands after failures. In contrast, in this chapter we investigate lightpath routing that maximizes the demand satisfaction of the logical graph after failures as well as lightpath routing that minimizes spare capacity requirements on the physical links that guarantees strong survivability as defined in Section 6.1.

6.3 Problem Description and Notations

First we define *weak survivability* and *strong survivability* in capacitated IP-over-WDM networks.

Definition 6.1 *An IP-over-WDM network with logical and physical topologies* $G_L = (V_L, E_L), G_P = (V_P, E_P)$ *is weakly survivable if after any physical link failure,* G_L *remains connected.*

Definition 6.2 *An IP-over-WDM network with* $G_L = (V_L, E_L), G_P = (V_P, E_P)$*, capacity* c_{ij} *for each physical link* (i, j) *and demand* d_{st} *for each logical link* (s, t) *is* *strongly survivable if after any physical link (i, j) failure, G_L remains connected and d_{st} can be satisfied for all $(s, t) \in E_L$.*

Definition 6.3 *The spare capacity on a physical link is the extra capacity required to satisfy all d_{st} after any (i, j) failure while the logical topology remains connected.*

Note: If the spare capacity requirement on each physical link is zero after a physical link failure, then the network is strongly survivable.

We will propose mathematical programming formulations for the following problems:

Problem 6.1 *Determine a lightpath routing that guarantees weak survivability and the logical link demand satisfaction after a physical link failure.*

Problem 6.2 *Determine a lightpath routing that guarantees strong survivability under minimum spare capacity requirements.*

The necessary and sufficient condition for weakly survivability is given in [5] and described in Section 2.2. We present in another form the necessary and sufficient condition for the survivable routing in the IP-over-WDM network after any physical link failure without considering logical demand and physical demand. We claim that after any physical link failure, if the logical network is connected, then, there exists at least one corresponding spanning tree to connect every nodes in the logical network. We let $\mathcal{T}_L$ be a spanning tree in the logical network and $\mathcal{S}_{ij}$ be a logical link set whose elements are routed through physical link (i, j). We now have the following.

Proposition 6.1 *A routing is survivable if and only if after any physical link (i, j) fails, there exists a spanning tree $\mathcal{T}_L \in G_L$ and*

$$\mathcal{T}_L \cap \mathcal{S}_{ij} = \emptyset. \tag{6.1}$$

Theorem 6.1 *The weakly survivable routing and strongly survivable routing design problem is NP-complete.*

The proof of Theorem 6.1 is similar to the proof of Theorem 2 in [5].

To tackle the problems proposed, we follow a two-stage design approach. In the first stage we determine a lightpath routing that guarantees weak survivability against any physical link failure. In the second stage, we consider the demand satisfaction in the logical network for the two problems we proposed.

6.3.1 Weakly Survivable Routing and Maximizing Routed Logical Link Demands

In the section we investigate Problem 6.1, namely, lightpath routing that guarantees weak survivability after any physical link failure. Towards this goal we proceed in two stages.

Stage 1: We design the IP-over-WDM network such that the logical topology remains connected after any physical link failure with the objective of maximizing logical demand satisfaction (logical demands routable under the selected lightpath routing) after any physical link failure.

Stage 2: With the information of existing lightpaths and the physical link failure, the demands/flow on the failed lightpaths need to be rerouted and the objective is to minimize the maximal unsatisfied demands caused by each physical link failure.

Next we describe an MILP formulation of the stage 1 of Problem 6.1. The first stage constraints provide lightpath routing for each logical demand that satisfies physical link capacity constraints and keeps the logical network connected after any physical link failure. Note here that a logical link representing connectivity between nodes s and t will be denoted by (s, t) if $s < t$, otherwise by (t, s). We consider logical links in the logical link collection, L_L. For stage 1 problem, we let $L_L = E_L$.

The constraints and optimization objective of the first stage of Problem 6.1 are given in Table 6.1. We formulate the first stage constraints as follows.

Parameter	Description	Info.
c_{ij}	capacity on the physical link (i, j)	given.
d_{st}	demand for the logical link (s, t)	given.
g_{st}	indicator for the logical link (s, t) whether (s, t) is connected or not	given
Variable	Description	Info.
h_{st}	the indicator whether the logical link (s, t) is connected	first stage for Problem 6.1.
y_{ij}^{st}	binary variable indicates whether the logical link $(s, t) \in E_L$ is routed through the physical link $(i, j) \in E_P$. If yes, $y_{ij}^{st} = 1$, otherwise, $y_{ij}^{st} = 0$.	first stage for Problem 6.1.
f_{ij}^{st}	flow on physical link (i, j) due to lightpath (s, t)	first stage for Problem 6.1.
r_{st}^{ij}	fractional variable for connectivity constraints.	first stage for Problem 6.1.
ρ_{st}	the capacity for the logical link (s, t), where ρ_{st} is the smallest capacity of links in the lightpath.	first stage for Problem 6.1.
θ	variable for max single logical link capacity.	first stage for Problem 6.1.
u_{ij}	link utilization request on the physical link (i, j)	given.
λ_{ij}^{st}	maximal flow for logical link (s, t) after a physical link failure and re-routing	second stage for Problem 6.1.
$x_{k\ell ij}^{st}$	rerouted flow on (k, ℓ) which can be maintained after the physical link (i, j) failure and re-routing.	second stage for Problem 6.1.
$z_{k\ell ij}^{st}$	binary variable indicates whether (s, t) is rerouted through (k, ℓ) after (i, j) failure.	second stage for Problem 6.1.
η_{ij}	amount of spare capacity required on the physical link (i, j) to satisfy strong survivability	second stage of Problem 6.2.

Table 6.1: Variables used in MILP formulation

Lightpath constraint:

$$\sum_{(i,j)\in E_P} y_{ij}^{st} - \sum_{(j,i)\in E_P} y_{ji}^{st} = 1 \qquad \text{if } s = i, (s,t) \in L_L \tag{6.2}$$

$$\sum_{(i,j)\in E_P} y_{ij}^{st} - \sum_{(j,i)\in E_P} y_{ji}^{st} = -1 \qquad \text{if } t = i, (s,t) \in L_L \tag{6.3}$$

$$\sum_{(i,j)\in E_P} y_{ij}^{st} - \sum_{(j,i)\in E_P} y_{ji}^{st} = 0 \qquad \text{otherwise}, (s,t) \in L_L \tag{6.4}$$

$$y_{ij}^{st} + y_{ji}^{st} \leq 1 \qquad (s,t) \in L_L, (i,j) \in E_P \tag{6.5}$$

Flow equivalence constraint:

$$f_{k\ell}^{st} + M(y_{k\ell}^{st} - 1) \leq f_{pq}^{st} + M(1 - y_{pq}^{st}), \quad (s,t) \in L_L, (k,\ell), (p,q) \in E_P, (k,\ell) \neq (p,q) \tag{6.6}$$

Flow conservation constraint:

$$\sum_{(i,j)\in E_P} f_{ij}^{st} - \sum_{(j,i)\in E_P} f_{ji}^{st} = \rho_{st} \qquad \text{if } s = i, (s,t) \in L_L \tag{6.7}$$

$$\sum_{(i,j)\in E_P} f_{ij}^{st} - \sum_{(j,i)\in E_P} f_{ji}^{st} = -\rho_{st} \qquad \text{if } t = i, (s,t) \in L_L \tag{6.8}$$

$$\sum_{(i,j)\in E_P} f_{ij}^{st} - \sum_{(j,i)\in E_P} f_{ji}^{st} = 0 \qquad \text{otherwise}, (s,t) \in L_L \tag{6.9}$$

$$\rho_{st} \leq d_{st} \qquad (s,t) \in L_L \tag{6.10}$$

Bounded flow constraint:

$$(f_{ij}^{st} + f_{ji}^{st}) \leq M y_{ij}^{st} \qquad (s,t) \in L_L, (i,j) \in E_P \tag{6.11}$$

Capacity constraint:

$$\sum_{(s,t)\in L_L} (f_{ij}^{st} + f_{ji}^{st}) \leq c_{ij} \qquad (i,j) \in E_P \tag{6.12}$$

Survivability constraint:

$$\sum_{(s,t)\in L_L} r_{st}^{ij} - \sum_{(t,s)\in L_L} r_{ts}^{ij} = -1 \qquad \text{if } s = v_1, (i,j) \in E_P \tag{6.13}$$

$$\sum_{(s,t)\in L_L} r_{st}^{ij} - \sum_{(t,s)\in L_L} r_{ts}^{ij} = \frac{1}{|V_L| - 1} \qquad \text{otherwise } (i,j) \in E_P \tag{6.14}$$

$$0 \leq r_{st}^{ij} \leq 1 - (y_{ij}^{st} + y_{ji}^{st}) \qquad (s,t) \in L_L, (i,j) \in E_P \tag{6.15}$$

$$0 \leq r_{ts}^{ij} \leq 1 - (y_{ij}^{st} + y_{ji}^{st}) \qquad (s,t) \in L_L, (i,j) \in E_P \tag{6.16}$$

Congestion constraint (optional):

$$\sum_{(s,t)\in L_L} (y_{ij}^{st} + y_{ji}^{st}) \leq u_{ij} \qquad (i,j) \in E_P \tag{6.17}$$

WSRD-CC Algorithm (First stage of Problem 6.1) – MILP formulation for the weakly survivable routing design (objective: maximize total logical link demands):

$$\textbf{WSRD-CC:} \ \max \sum_{(s,t)\in L_L} \rho_{st} \tag{6.18}$$

subject to: Constraint (6.2) to (6.16),

$$y_{ij}^{st} \in \{0,1\}, r_{ij}^{st} \geq 0, f_{ij}^{st} \geq 0, \rho_{st} \geq 0, \quad (s,t) \in L_L, (i,j) \in E_P \tag{6.19}$$

MILP formulation for the weakly survivable routing (First stage of Problem 6.1) (objective: maximize the minimal demand satisfaction on a single logical link):

$$\max\ \theta$$
$$\text{where}\ \ \theta \leq \rho_{st}. \tag{6.20}$$

Next, we explain the purpose of each constraints.

Proposition 6.2 *The lightpath constraints provide lightpaths for $(s, t) \in G_L$ and eliminate cycles, which are routed through both (i, j) and (j, i).*

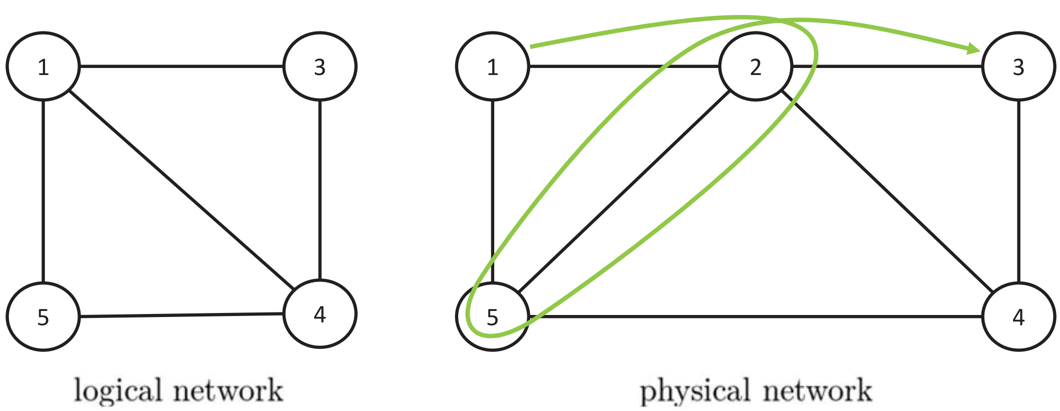


Figure 6.2: Example of a lightpath routing through both direction of (i, j)

Constraints (6.2) – (6.4) guarantee a single lightpath for each logical link (s, t). This is achieved by requiring the binary decision variables y_{ij}^{st} to satisfy the flow constraints. $y_{ij}^{st} = 1$ defines a single lightpath for logical link (s, t) routed through physical links (i, j). However, the lightpath for logical link (s, t) may contain cycle(s). We demonstrate the existence of a cycle in the lightpath for logical link (s, t) which satisfies constraints (6.2) – (6.4) in Fig. 6.2. The lightpath for logical link $(1, 3)$ has physical links $(1, 2), (2, 5), (5, 2)$, and $(2, 3)$, which is routed through both $(2, 5)$ and $(5, 2)$.

Thus, besides the network flow conservation constraints, we add constraint (6.5) to eliminate cycle(s) such that the above situation would not exist.

Proposition 6.3 *Constraints* (6.2) – (6.5) *provide tighter feasible region for the survivable routing in the IP-over-WDM network than constraints* (6.2) – (6.4).

Proof: We consider a lightpath $\mathcal{P}_1$ for logical link (s,t) satisfying constraints (6.2) – (6.4) which does not contain any cycle. And we construct another lightpath, $\mathcal{P}_2$ which keeps a similar routing as $\mathcal{P}_1$ but contains a cycle, which is routed through both (i,j) and (j,i) arcs with $i \in \mathcal{P}_1$. Both $\mathcal{P}_1$ and $\mathcal{P}_2$ are feasible for constraints (6.2) – (6.4) and provide lightpaths for (s,t). After (i,j) failure, lightpath $\mathcal{P}_2$ is disconnected, but $\mathcal{P}_1$ is connected. Constraint (6.5) helps to rule out cases such as lightpath $\mathcal{P}_2$. Hence the conclusion holds. □

Proposition 6.4 *The flow equivalence constraint* (6.6) *forces flows to be the same for all the physical links on the lightpath selected for the demand d_{st} on link (s,t).*

Proof: We prove this proposition by considering three cases: (I) both (k,ℓ) and (p,q) are in the lightpath (s,t), (II) one of (k,ℓ) and (p,q) is in the lightpath (s,t), and (III) none of (k,ℓ) and (p,q) is in the lightpath (s,t).

For case I, both (k,ℓ) and (p,q) are in the lightpath (s,t). Then, both $y_{k\ell}^{st}$ and y_{pq}^{st} are equal to 1. Therefore, this constraint forces $f_{k\ell}^{st}$ and f_{pq}^{st} to be equal for every pair of links (k,ℓ) and (p,q) in the lightpath (s,t).

For case II, one of (k,ℓ) and (p,q) is in the lightpath for (s,t). Then, one of $y_{k\ell}^{st}$ and y_{pq}^{st} equals 1. If $y_{k\ell}^{st} = 1$ and $y_{pq}^{st} = 0$, then $f_{k\ell}^{st} \leq M$ because $f_{pq}^{st} = 0$. If $y_{k\ell}^{st} = 0$ and $y_{pq}^{st} = 1$, then $f_{k\ell}^{st} = 0$ and $f_{pq}^{st} \geq 0$. Thus, this constraint holds.

For case III, none of (k,ℓ) and (p,q) is in the lightpath for (s,t). Then, both $f_{k\ell}^{st}$ and f_{pq}^{st} are 0 due to $y_{k\ell}^{st} = y_{pq}^{st} = 0$. Thus this constraints holds.

Thus we have shown that constraint (6.6) guarantees the flows to be the same for all the physical links on the lightpath selected for routing the demand on link (s, t). □

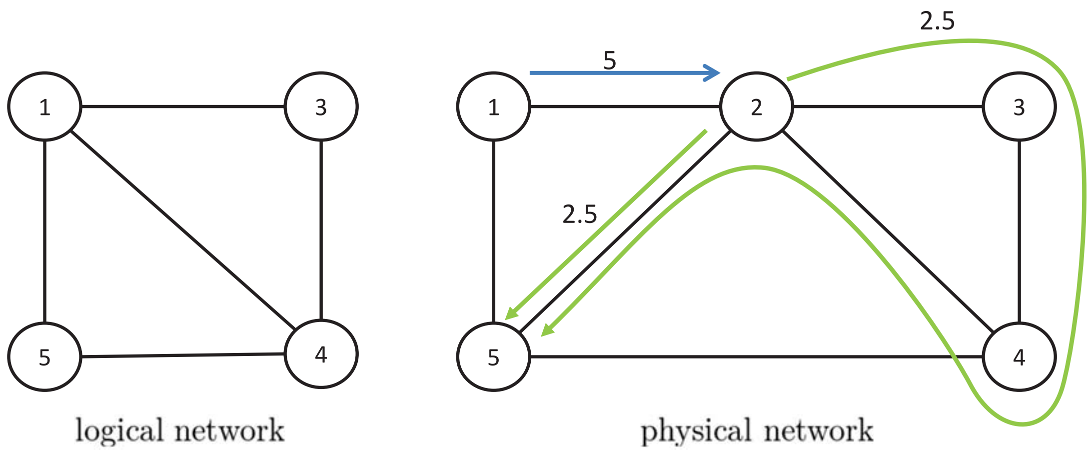


Figure 6.3: Example of necessity of flow equivalence constraints

We demonstrate the necessity of flow equivalence constraint as follows. Flow conservation constraints (6.7) – (6.10) require flows on links selected for the lightpath of logical link (s, t) to be less than or equal to d_{st}. But flow conservation constraints cannot restrict the flow on each physical link in the lightpath to be equal. We show it by an example in Fig. 6.3. The lightpath of logical link $(1, 2)$ including $(2, 3)$, $(3, 4)$, $(4, 2)$, and $(2, 5)$ satisfies all flow constraints, but the flow splits at node 2. Therefore, the flow on each physical link in the lightpath is not equivalent, which disobeys the flow equivalent assumption for a lightpath.

Bounded flow constraint (6.11) guarantees that each physical link carries flow only if the lightpath(s) is routed through the physical link. Capacity constraints (6.12) requires that the total flow in each physical link due to all the lightpaths be no more than the corresponding link capacity.

The idea of the survivability constraints came from Deng et al. [25].

Proposition 6.5 *The survivability constraints provide the necessary and sufficient condition for the survivable routing in the IP-over-WDM network.*

Proof: Based on the lightpaths generated by constraints (6.2) – (6.5), after physical link (i, j) failure, the lightpath (s, t) is disconnected if $y_{ij}^{st} + y_{ji}^{st} = 1$, otherwise, the lightpath for (s, t) remains connectivity.

Based on Proposition 6.1, there exists at least one spanning tree embedded in the logical topology after any physical link (i, j) failure. Hence, constraints (6.15) and (6.16) provides the information that whether (s, t) is connected after (i, j) failure. Instead of using constraints to eliminate cycles and construct a undirected spanning tree, we formulate constraints (6.13) – (6.16), which provide a directed tree in the logical network after (i, j) failure. Therefore, if there exists a feasible solution for (6.13) – (6.16), then, the logical network remains connected after physical link (i, j) failure.□

Congestion constraint (6.17) is used if the number of wavelength is limited on each fiber. We do not consider wavelength conversion and the number of available wavelengths for the problems discussed in this chapter.

With above constraints, the MILP formulation for the first stage of Problem 6.1 is in (6.2) – (6.16).

There are different ways to evaluate the largest capacity on the logical links. (6.18) requires maximization of the total capacity on the logical network. We also can maximize the largest capacity on the single logical link by maximizing the minimum capacity on the logical link as in (6.20).

From the first stage of Problem 6.1 (WSRD-CC algorithm) we obtain the lightpath routing information with the optimal solution y^*. We consider the second stage of this network design with respect to y^* and ρ^*.

Once a physical link (i,j) fails, we need to re-route lightpaths that were routed through link (i,j) to satisfy at least partially original demands on these lightpaths. With y^*, we know that if $y_{ij}^{st*} = 1$, then lightpath $s-t$ is routed through (i,j). Thus, for a given (i,j), we only need to re-route lightpaths that are in the set $R_{ij} = \{(s,t) : y_{ij}^{st*} = 1, (s,t) \in L_L\}$. Therefore, in the second stage, after any physical link (i,j) failure, the disrupted network flow is re-routed through a new lightpath going through physical links with enough residual capacities (the residual capacity on physical links after any physical link failure). The existence of the new lightpath routing is restricted by the residual capacities. We formulate the second stage constraints as follows:

Re-routing constraint (6.21) – (6.23) provide the new lightpaths for logical links which are broken after the (i,j) failure. The demands on the physical links that lie on these new lightpaths must be within their residual capacities.

The residual capacity of the physical link (k,ℓ) is $c_{k\ell} - \sum_{(u,v)\in E_L \setminus R_{ij}} \rho_{uv}^* y_{k\ell}^{uv}$.

Re-routing constraint: For all $(i,j) \in E_P$,

$$\sum_{(k,\ell)\in E_P\setminus\{(i,j)\}} z_{k\ell ij}^{st} - \sum_{(\ell,k)\in E_P\setminus\{(i,j)\}} z_{\ell kij}^{st} = 1 \qquad \text{if } s = k, (s,t) \in R_{ij} \tag{6.21}$$

$$\sum_{(k,\ell)\in E_P\setminus\{(i,j)\}} z_{k\ell ij}^{st} - \sum_{(\ell,k)\in E_P\setminus\{(i,j)\}} z_{\ell kij}^{st} = -1 \qquad \text{if } t = k, (s,t) \in R_{ij} \tag{6.22}$$

$$\sum_{(k,\ell)\in E_P\setminus\{(i,j)\}} z_{k\ell ij}^{st} - \sum_{(\ell,k)\in E_P\setminus\{(i,j)\}} z_{\ell kij}^{st} = 0 \qquad \text{otherwise}, (s,t) \in R_{ij}. \tag{6.23}$$

Flow equivalence constraint:

$$x_{k\ell ij}^{st} + M(z_{k\ell ij}^{st} - 1) \le x_{pqij}^{st} + M(1 - z_{pqij}^{st}), \tag{6.24}$$

$$(s,t) \in R_{ij}, (k,\ell), (p,q) \in E_P \setminus \{(i,j)\}.$$

Residual capacity constraint: For all $(i,j) \in E_P$,

$$\sum_{(s,t)\in R_{ij}} (x^{st}_{k\ell ij} + x^{st}_{\ell kij}) \le c_{k\ell} \quad - \sum_{(u,v)\in E_L\setminus R_{ij}} \rho^{*}_{uv} y^{uv*}_{k\ell}, \tag{6.25}$$

$$(k,\ell) \in E_P \setminus \{(i,j)\}$$

$$x^{st}_{k\ell ij} \le M z^{st}_{k\ell ij}, \qquad (s,t) \in R_{ij}, (k,\ell) \in E_P \setminus \{(i,j)\} \tag{6.26}$$

$$\lambda^{st}_{ij} \ge x^{st}_{k\ell ij}, \qquad (s,t) \in R_{ij}, (k,\ell) \in E_P \setminus \{(i,j)\} \tag{6.27}$$

$$\lambda^{st}_{ij} \le x^{st}_{k\ell ij} + M(1 - z^{st}_{k\ell ij}), \qquad (s,t) \in R_{ij}, (k,\ell) \in E_P \setminus \{(i,j)\}. \tag{6.28}$$

Algorithm MAXCAP-WSRD (Second stage of Problem 6.1) MILP formulation for the weak survivability design (objective: max demand satisfaction)

$$\max \sum_{(s,t)\in E_L} \sum_{(i,j)\in E_P} \lambda^{st}_{ij} \tag{6.29}$$

$$\text{subject to: Constraints (6.21) to (6.24)}$$

$$z^{st}_{k\ell ij} \in \{0,1\}, \lambda^{st}_{ij}, x^{st}_{k\ell ij} \ge 0$$

$$(i,j) \in E_P, (s,t) \in L_L, (k,\ell) \in E_P \setminus \{(i,j)\} \tag{6.30}$$

Residual capacity constraint (6.25) restricts the total rerouted flow on link (k,ℓ) to be within its residual capacity.

Constraints (6.25) – (6.28) guarantee that the demand d_{st} rerouted along the lightpath for a broken logical link (s,t) due to the failure of physical link (i,j) is equal to the flows on the links of the lightpath. Constraint (6.26) restricts the flow on link (k,ℓ) due to the rerouting of the disrupted flow for (s,t) after the physical link (i,j) failure. Here M is a large number greater than the maximum link capacity.

The goal for the second stage of Problem 6.1 is to minimize the total unsatisfied demand, or equivalently, maximize the total fulfilled demand in the capacitated IP-over-WDM network by appropriately rerouting after a failure occurs. The MILP formulation of the second stage of Problem 6.1 (called algorithm MAXCAP-WSRD) is listed in (6.21) – (6.30).

In the next section, we discuss the spare capacity allocation to achieve strong survivability.

6.3.2 Strongly Survivable Lightpath Routing under Minimum Physical Spare Capacity

In order to satisfy all logical demands, we study spare capacity allocation. There are two ways to allocate spare capacities: spare capacity allocation on the logical link, and spare capacity allocation on the physical link. Sahasrabuddhe et al. in [14] demonstrated the protection at the WDM layer, (i.e., set up a backup lightpath for every primary lightpath and the corresponding maximum capacity allocation. Chu et al. in [51] and Zhang and Durresi in [52] also considered spare capacity allocation at the WDM layer. Kan et al. in [22] presented spare capacity allocation on logical links.

Compared with spare capacity allocation in the logical network, the spare capacity allocation in the physical network reflects real insufficient capacity which restricted demand satisfaction, "since capacity of IP layer is carried by WDM layer" [52]. In the following, we present the spare capacity allocation on the physical links to satisfy all demands from logical network.

We investigate Problem 6.2 which requires the design of a strongly survivable lightpath routing that does not violate physical link capacity requirements. While doing so, we may have to add additional capacity (called spare capacity) to some of the physical links so that all the logical demands can be fully routed. Our objective is to minimize the total spare capacity added.

Towards the above objective we proceed in two stages.

Stage 1: We determine a weakly survivable lightpath routing. Note that such a routing ensures that the logical network remains connected after any physical link failure.

Stage 2: Add spare capacity to the physical links and re-route the flows on logical links that are broken due to a physical link failure. This is to ensure that all the logical demands are satisfied after a physical link failure. Our objective is to minimize the total spare capacity required.

The MILP formulation for the design of a strongly survivable routing requiring minimum spare capacities for rerouting the failed logical demands as follows:

Stage 1:

$$\max \sum_{(s,t)\in E_L} \rho_{st} \tag{6.31}$$

$$\text{subject to: Constraint (6.2) to (6.16)} \tag{6.32}$$

Stage 2:

$$\min \sum_{(i,j)\in E_P} \eta_{ij} \tag{6.33}$$

$$\text{subject to:} \quad \text{Constraints (6.21) to (6.26)},$$

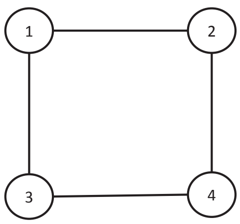
$$\sum_{(s,t)\in R_{ij}} d_{st}(z^{st}_{k\ell ij} + z^{st}_{\ell kij}) \le c_{k\ell} - \sum_{(s,t)\in E_L \setminus R_{ij}} \rho^*_{st} y^{st*}_{k\ell} + \eta_{k\ell}, \tag{6.34}$$

$$(i,j) \in E_P, (k,\ell) \in E_P \setminus (i,j).$$

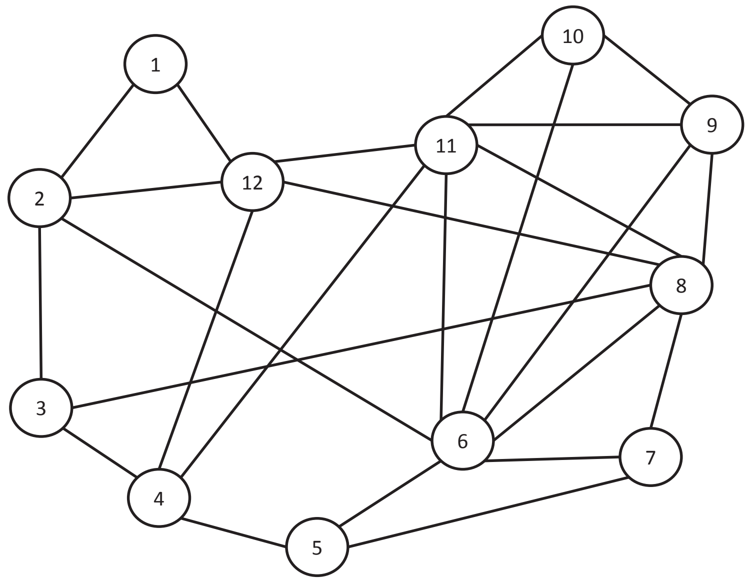
6.4 Simulation and Experimental Evaluation

In this section we report our results on the effectiveness of our formulations. We used CPLEX 12.1 to run the weakly and strongly survivable MILP formulations. We adopted the networks introduced in [53] as physical topologies (Networks 2 and 7 (European 1 and 2), and Networks 3 and 6). We also generated a SMALL network with

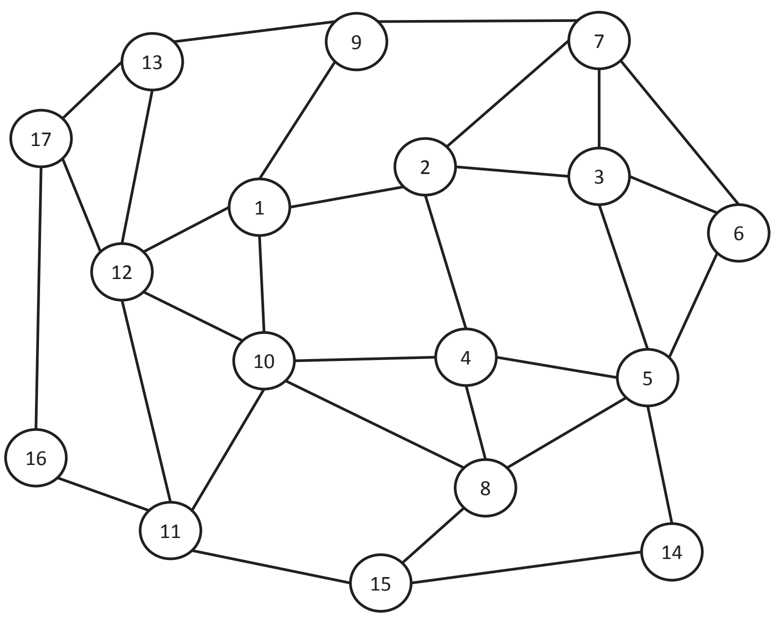
4(3) nodes and RAND (random network) with 25(12) nodes in the physical(logical) topology. Corresponding logical topologies are chosen to be two-connected. Logical nodes are subsets of physical nodes, that is, $|V_L| = 0.5 * |V_P|$.



(a) SMALL

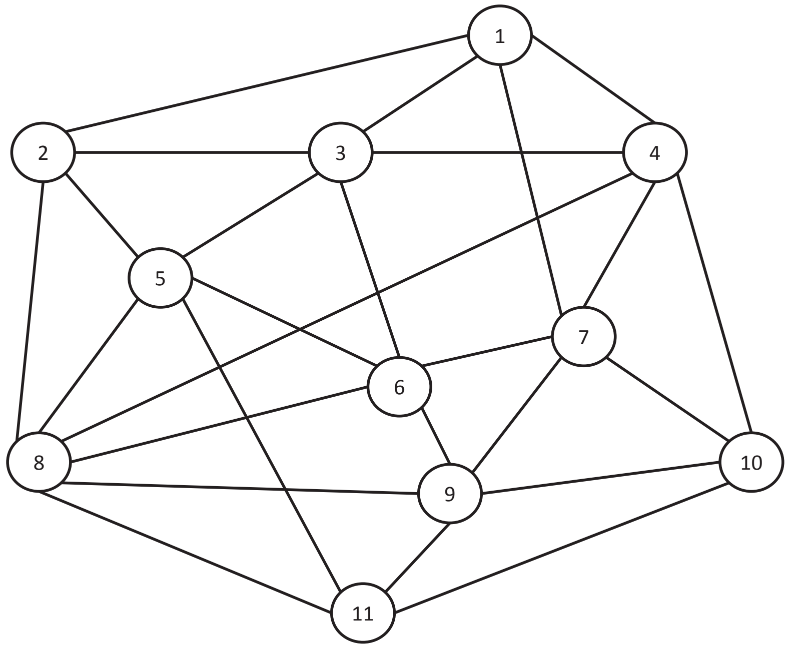


(b) G3

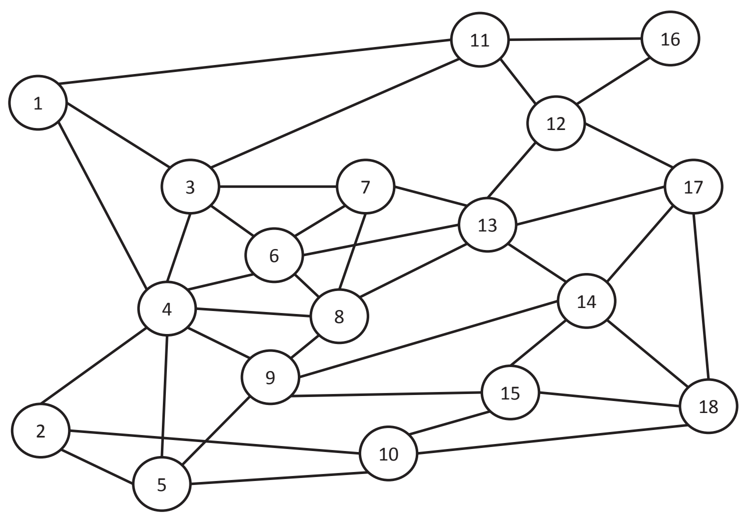


(c) G6

Figure 6.4: (a) Network SMALL; (b) Network 3 (G3); (c) Network 6 (G6)



(a) EURO 1



(b) EURO 2

Figure 6.5: (a) European Network (EURO 1); (b) European Network (EURO 2)

We report results of our experiments on the topologies shown in Tables 6.2 and 6.3. As expected, MILP formulations require high execution times, though they give optimum values of the required results. We compared the results of MILP formulations with certain heuristics that take comparatively smaller execution times. The heuristics were implemented using LEMON library [45].

A brief outline of the methods used for the heuristics are discussed next.

Stage 1 of Problem 6.1:

The heuristic algorithm for the weakly survivable routing problem is as follows. The datum node is denoted as Δ. Pick any logical node v with degree ≥ 2 and map two of v's adjacent edges into disjoint paths in the physical topology, and then remove v and all its adjacent nodes from the logical topology. This procedure is repeated until no logical nodes with degree ≥ 2 is left. Next, pick a node v from the remaining logical topology with degree $= 1$ (an edge (u, v)), add a parallel edge $(u, v)'$ to (u, v) and then find disjoint mappings for (u, v) and $(u, v)'$ in the physical topology. This procedure is executed until all logical nodes with degree $= 1$ are eliminated. If after the previous steps, there exist nodes v with degree $= 0$, add two parallel edges connecting v and Δ and map them disjointly in the physical topology. The augmented logical topology is denoted as $\mathbb{L}$.

The above procedures generate a survivable routing for the augmented logical topology. Proof of correctness of this may be found in Chapter 4 and [37].

We next push a flow of value d_{st} along the lightpath that corresponds to the logical link (s, t) for each $(s, t) \in \mathbb{L}$. The physical link capacities required to satisfy the specified logical demands d_{st} are used as the given physical link capacities. Thus at the end of the first stage we will have a logical topology that has a survivable lightpath routing and physical link capacities that accommodate all the logical demands.

Stage 2 of Problem 6.1:

In this stage we take down each physical link (i, j) (representing the link failure) one at a time. The lightpaths (the corresponding logical links) that use this link will be broken. Let R_{ij} be the set of logical links that are broken due to the failure of physical link (i, j). We then calculate the residual capacity available on each physical link after the failure of (i, j). For each logical link in R_{ij} we find a new lightpath that avoids the physical link (i, j). We choose a path in the physical topology with the largest residual capacity as the new lightpath. If the logical demand can be satisfied by the new lightpath, the demand is subtracted from the capacity of the links on the lightpath and this demand (s, t) is marked as fully satisfied after failure. Otherwise, we calculate the largest possible demand which can be satisfied and push that as flow on the lightpath and subtract it from the capacity of physical links on the lightpath. Every time we calculate this new logical demand, we also recalculate the residual capacities on the physical links in the selected lightpath.

Stage 1 of Problem 6.2 is the same as the stage 1 in the case of Problem 6.1.

Stage 2 of Problem 6.2:

In this stage we can take down each physical link (i, j) (representing the link failure) one at a time. Using the information about the lightpaths generated in stage 1, we calculate the residual capacities available on the physical links after the failure of (i, j). For each failed logical link (s, t) we find a new lightpath that avoids the physical link (i, j). We choose the new lightpath which has the maximum residual capacity and record the extra capacities required on the physical links to satisfy d_{st} if d_{st} is larger than the residual capacity on the chosen lightpath.

The results of these heuristics are compared with the result of the MILP formulations as in Tables 6.2 – 6.3. Table 6.2 compares the total demands satisfied in each case after a physical link failure. The two values, for example, 69/71 in the MILP result of Network 6 (G6), denote that 69 out of 71 affected demands can be satisfied. Notice that the number of total affected demands are different for MILP and heuristic results because the lightpath routings generated are different. From the result we can see that different lightpath routes for the MILP and the heuristic have a strong impact on the satisfied demands after failure. The trade-off between the MILP and heuristic approach is that the computation time for the heuristic is about 50 times less than that for the MILP even on a physical topology with a few dozen nodes, while at the same time the heuristic provides a result which is close to the optimal solution.

Table 6.3 compares the minimum spare capacity required by applying MILP and heuristic approaches. From the table we can see that our heuristic for the strongly survivable case can actually provides a result very close to the optimal solution (or even the optimal solution).

	SMALL	G3	G6	EURO 1	EURO 2	RAND
MILP	0/28	38/38	69/71	17/17	45/47	361/387
Ratio	0%	100%	97%	100%	96%	93%
Heuristic	0/28	32/36	44/62	10/17	41/47	317/372
Ratio	0%	89%	71%	59%	87%	85%

Table 6.2: Comparison of MILP and heuristic results on demand satisfaction after failure (weakly survivable)

	SMALL	G3	G6	EURO 1	EURO 2	RAND
MILP	26	3	17	10	6	2
Heuristic	26	3	21	12	6	18

Table 6.3: Comparison of MILP and heuristic results on minimum spare capacity (strongly survivable)

6.5 Extensions of Weakly Survivable Routing Problem

In this section, we discuss extensions of the weakly survivable routing formulations in an given IP-over-WDM network from the following three perspectives: (1) augmentation of the logical network to guarantee a survivable routing, (2) logical demand load balancing, and (3) survivable routing with multiple physical links failures.

6.5.1 Augmentation in the Logical Network

For an IP-over-WDM network, it is possible that WSRD-CC algorithm does not have a feasible solution. The MILP formulation only provides survivable routing in the IP-over-WDM network, when the MILP has a feasible solution, which implies that the survivable routing exists in the given IP-over-WDM network.

To the best of our knowledge, there is no result about how to guarantee the existence of a survivable routing in a given two-layer network . With two-node connectivity in the physical network, if we allow augmentation on the logical network, then, we can generate the survivable routing for the IP-over-WDM network with augmented logical network. Note that this augmentation was considered in Chapter 4 for the case of uncapacitated networks. Here we provide an exact MILP formulation to achieve augmentation for guaranteed survivability in a capacitated network.

Let F_L be the set of all pairs of nodes in G_L. That is

$$F_L = \{(i, j)|i, j \in V_L\}.$$

Let $h_{s,t}$ be a new variable that indicates whether (s, t) is to be used for augmentation. To guarantee that all original links in E_L are also included in the augmented network, we set

$$g_{st} \leq h_{st}, \qquad (s, t) \in F_L \tag{6.35}$$

$$y_{ij}^{st} \leq h_{st}, \qquad (s, t) \in F_L, (i, j) \in E_P. \tag{6.36}$$

Constraint (6.36) guarantees that the lightpath is generated only if (s, t) is in the augmented network. After replacing $L_L = E_L$ in WSRD-CC algorithm by $L_L = F_L$, with constraints (6.35) and (6.36), we obtain the mixed-integer formulation for the survivable routing in given IP-over-WDM network with augmentation. In order to reduce the size of the MILP formulation, we also could restrict logical link augmentation to a smaller sub-network of the logical network.

6.5.2 Load Balancing

In the WSRD-CC algorithm, with the consideration of the maximum demand satisfaction, it could happen that several logical demands are satisfied, but the rest of them are not satisfied at all. For the IP-over-WDM network, the logical demand reflects customers' requests. Hence, the demand satisfaction with fairness is important. This motivates us to consider balancing of logical demands.

Definition 6.4 *Load balancing with absolute fairness requires that all logical demands satisfied achieve a ratio,* Ω*, of the original demands.*

Definition 6.5 *Load balancing with ratio-weighted fairness requires that the logical demand on every logical link* (s, t) *achieves a ratio,* β_{st}*, of its demand.*

We present an exact solution approach to achieve the logical load balancing with absolute fairness and a $1 - \varepsilon$ approximation algorithm for the logical load balancing with ratio-weighted fairness.

Load Balancing with Absolute Fairness

We present an ILP formulation for the exact solution for logical load balancing with absolute fairness. The mixed-integer program for the load balancing with absolute fairness is as follows:

(WSRD-AF)

$$\max \ \Omega \tag{6.37}$$

subject to: Constraints (6.2) to (6.6) and Constraints (6.9) to (6.16),

$$\sum_{(i,j)\in E_P} f_{ij}^{st} - \sum_{(j,i)\in E_P} f_{ji}^{st} = \Omega d_{st} \qquad \text{if } s = i, (s,t) \in L_L \tag{6.38}$$

$$\sum_{(i,j)\in E_P} f_{ij}^{st} - \sum_{(j,i)\in E_P} f_{ji}^{st} = -\Omega d_{st} \qquad \text{if } t = i, (s,t) \in L_L \tag{6.39}$$

$$0 \leq \Omega \leq 1. \tag{6.40}$$

The objective function (6.37) considers the maximum concurrent logical demand flow via lightpaths, which aims to assign logical demand through each lightpath, such that the ratio between demand satisfied and the original demand is the same for all logical links. Hence, the maximum logical demand satisfaction for each logical node pair is absolutely fair.

A simpler (not necessarily exact) approach to achieve load balancing with absolute fairness, assuming that we are given a survivable routing (without consideration of capacity constraints) is given next. Let $r(s,t)$ be the lightpath for the logical link (s,t). Then the following linear program will achieve load balancing with absolute fairness, for a given survivable routing.

(ABF1)

$$\max_{\Omega,\rho} \Omega$$

subject to: $$\rho_{st} = \Omega d_{st}, \qquad (s,t) \in E_L \tag{6.41}$$

$$\sum_{\{(s,t)\in E_L, (i,j)\in r(s,t)\}} \rho_{st} \leq c_{ij}, \qquad (i,j) \in E_P \tag{6.42}$$

$$0 \leq \Omega \leq 1 \tag{6.43}$$

Constraints (6.41) and (6.42) provide the demand and capacity bounds of the maximum concurrent logical demand flow based on the logical node pair and the physical edge through which the lightpath routes.

Constraints (6.41) and (6.42) can be restated as

$$\Omega \sum_{(s,t)\in E_L,(i,j)\in r(s,t)} d_{st} \le c_{ij}, \qquad (i,j) \in E_P. \tag{6.44}$$

Then, we have

(ABF)

$$\max_{\Omega,\rho} \Omega$$
$$\text{subject to: Constraints (6.43) and (6.44).} \tag{6.45}$$

Based on (ABF), for a given survivable routing we present the exact solution approach for the logical capacity balance with absolute fairness as follows:

Step 1. Calculate total logical demand request which expects to be routed through physical edge $(i,j) \in E_P$, D_{ij}: $D_{ij} = \sum_{(s,t)\in E_L,(i,j)\in r(s,t)} d_{st}$.

Step 2. Calculate the maximum concurrent logical demand flow Ω, $\Omega = \min\{\min\{c_{ij}/n_{ij}, (i,j) \in E_P\}, 1\}$, $n_{ij} = \sum_{(s,t)\in E_L,(i,j)\in r(s,t)} d_{st}$.

Load Balancing with Ratio-weighted Fairness

Next we present an MILP formulation and construct an approximation algorithm for load balancing with ratio weighted fairness. We realize the ratio-related fairness by the construction of a piecewise linear objective function.

We define the evaluation function $f(\cdot)$ of ρ_{st},

$$f(\rho_{st}) = \begin{cases} \rho_{st} & \text{if } \rho_{st} \leq \beta_{st} d_{st}, \\ \beta_{st} d_{st} + \alpha_{st}(\rho_{st} - \beta_{st} d_{st}) & \text{if } \rho_{st} > \beta_{st} d_{st}. \end{cases} \tag{6.46}$$

Here $0 \leq \beta_{st} \leq 1$ and $0 \leq \alpha_{st} \leq 1$ are specified for each (s, t). The above evaluation function is a piecewise linear function. If demand satisfied is less than β_{st} portion of demand request between s and t, then, $f(\rho_{st})$ is ρ_{st}. If demand satisfied is more than $\beta_{st} d_{st}$, then, we use α_{st} (a penalty factor) to obtain $f(\rho_{st})$. The purpose of introducing this piecewise linear function is to put different weights on demand satisfactions.

The following MILP achieves a survivable routing with ratio-weighted load balancing.

(WSRD-RWF)

$$\max \sum_{(s,t) \in E_L} f(\rho_{st}) \tag{6.47}$$

subject to: Constraints (6.2) to (6.16)

$$y_{ij}^{st} \in \{0, 1\}, r_{ij}^{st} \geq 0, f_{ij}^{st} \geq 0, \rho_{st} \geq 0 \quad (i, j) \in E_P, (s, t) \in E_L \tag{6.48}$$

Objective function (6.47) encourages β_{st} portion of logical demand to be satisfied for each logical node pair (s, t). As before, we can provide a simpler approach if we assume that we already have a survivable routing in the IP-over-WDM network. We can then restate WSRD-RWF as follows:

(RRF)

$$\max \sum_{(s,t)\in E_L} f(\rho_{st}) \tag{6.49}$$

$$\text{subject to:} \quad \sum_{(i,j)\in r(s,t)} \rho_{st} \leq c_{ij}, \qquad (s,t)\in E_L, (i,j)\in E_P \tag{6.50}$$

$$\rho_{st} \leq d_{st}, \qquad (s,t)\in E_L \tag{6.51}$$

$$\rho_{st} \in Z^+, \qquad (s,t)\in E_L \tag{6.52}$$

Constraint (6.50) provides the capacity bound for all routings which are routed through physical edge (i,j). Constraint (6.51) restricts that the demand satisfied does not exceed logical demand between the start and end points of the routing and capacity on edge (i,j).

Formulation (RRF) is a concave objective function. Its maximal solution cannot be obtained at the boundary of the feasible region. Therefore, we next transform it to an equivalent integer linear program. Then, we present an approximation algorithm which is based on a linear program which is a relaxation of the integer linear program.

First, we transfer (RRF) to an equivalent integer linear programming problem by introducing two auxiliary variables μ and ν. We let μ_{st} represent the demand satisfaction which is lower than β_{st} portion of logical demand d_{st} between logical node pair (s,t) and ν_{st} represent the demand satisfaction which excess $\beta_{st} d_{st}$. Then, we obtain

Proposition 6.6 *The equivalent integer linear programming problem of (RRF) is as follows:*

(RFR)

$$\max \sum_{(s,t)\in E_L} \mu_{st} + \sum_{(s,t)\in E_L} \alpha_{st}\nu_{st} \tag{6.53}$$

$$\text{subject to:} \quad \sum_{(s,t)\in E_L,(i,j)\in r(s,t)} (\mu_{st} + \nu_{st}) \le c_{ij}, \qquad (i,j) \in E_P \tag{6.54}$$

$$\nu_{st}/(1-\beta_{st}) \le \mu_{st}/\beta_{st}, \qquad (s,t) \in E_L \tag{6.55}$$

$$\mu_{st} \le \beta_{st} d_{st} \qquad (s,t) \in E_L \tag{6.56}$$

$$\nu_{st} \le (1-\beta_{st}) d_{st} \qquad (s,t) \in E_L \tag{6.57}$$

$$\mu_{st}, \nu_{st} \in Z^+, \qquad (s,t) \in E_L \tag{6.58}$$

We now demonstrate why (RFR) is equivalent to (RRF). Constraints (6.56) and (6.57) put limits on two parts of demand satisfaction, μ_{st} and ν_{st}. Constraint (6.55) forces to satisfy β_{st} portion of d_{st}. Therefore, linear constraints and objective function provide the same information as the piecewise linear objective function.

Note here that constraint (6.55) could be removed from (RFR), because the maximal objective function forces μ to be larger which has larger weight 1 than α_{st} with $\alpha_{st} \le 1$.

In the following, we demonstrate a polynomial approximation algorithm for the logical capacity balance with ratio-weighted fairness which follows the approximation scheme for multidimensional $\{0, 1\}$ knapsack problem given in [54].

We present the detailed approximation algorithm for (RFR) as follows. First, we introduce the linear relaxation based heuristics for (RFR), which provide the basis for our approximation algorithm.

Linear Relaxation Based Heuristic for an Integer Program

Step 1. Solve the linear relaxation of an integer program $\max\{cx, Ax \le b, x \in Z_m^+\}$ and record solution $\widetilde{x}^*$.

Step 2. Round down the fractional solution to be integer solution, $\lfloor \widetilde{x}_i^* \rfloor$.

Step 3. Let $Z_\ell = \sum_{i=1}^{m} c_i \lfloor \widetilde{x}_i^* \rfloor$.

Based on the linear relaxation based heuristic, we generate the rounded integral solution for (RFR). Then, we present our approximation algorithm for logical capacity balance with ratio-related fairness.

An Approximation Algorithm for Logical Capacity Balance with Ratio-related Fairness

Step 1. Construct feasible region for μ and ν with parameter ε.

Step 1.1 Let $\delta = \lceil k(1/\varepsilon - 1) \rceil$, where $k = 2|E_L|$.

Step 1.2 Let $\Lambda = \{\sum_{(s,t)\in E_L}(\mu_{st} + \nu_{st}) \leq \delta$, and constraints (6.54), (6.56), (6.57), (6.58)$\}$ [Note: Find a feasible solution for (RFR)].

Step 2. Solve a corresponding linear program for each $(\overline{\mu}_{st}, \overline{\nu}_{st}) \in \Lambda$.

Step 2.1 Assume that the $\overline{\nu_{st}}$'s are arranged such that the $\overline{\nu_{st}}$ with higher value of α_{st} appears first. Then arrange $\overline{\mu_{st}}$ and $\overline{\nu_{st}}$ so that $\overline{\mu_{st}}$'s appear first and $\overline{\nu_{st}}$'s appear in the order specified above. Now let m be the largest index in the variables for which the corresponding variable in the solution constructed in step 1.2 is not equal to 0.

Step 2.2 Construct the following linear program (LR) where

(a) Z_i: represents an μ_{st} or ν_{st} variable

(b)

$$\begin{aligned} a_i &= 1 && \text{if } Z_i \text{ is a } \mu_{st} \text{ variable} \\ &= \alpha_{st} && \text{if } Z_i \text{ is a } \nu_{st} \text{ variable} \end{aligned}$$

(c) if i corresponds to (s,t) then $r(i)$ will denote $r(s,t)$,

$$\max \quad \sum_{i\geq m+1} a_i Z_i \tag{6.59}$$

$$\text{subject to:} \quad \sum_{(i,j)\in r(k), k\geq m+1} Z_k \leq c_{ij} - \sum_{\ell=1,\forall(i,j)\in E_P}^{2|E_L|} x_\ell. \tag{6.60}$$

Step 3. Generation of the best objective value of approximation algorithm: pick the best value formed in step 2.2.

We let Z_r^* denote the optimal solution of (RFR) and Z_a be the objective value of above approximation algorithm (calculated in step 3). The following theorem is a direct consequence of the analysis in [54].

Theorem 6.2 (1) *Approximation algorithm for logical capacity balance with ratio-related fairness takes time* $\mathcal{O}(|E_L|^{\lceil 1/\varepsilon \rceil})$ *and* $\mathcal{O}(|E_L|)$ *space.*
(2) *If the optimal solution of (RFR) is non-zero and finite, then,* $Z_a/Z_r^* > 1-\varepsilon$.

The approximation algorithm given above is computationally intractable because it requires to consider all feasible solutions. However, the algorithm could be the basis of an approximation heuristic. For example, instead of considering all feasible solutions, we can consider a subset of these solutions.

6.5.3 Survivable Routing with Multiple Failures

In this section, we extend our results for the survivable routing in an IP-over-WDM network with single physical link failure case to the multiple physical link failures scenario.

Kurant and Thiran defined in [31] the k-survivable of the IP-over-WDM network with k link failures as follows.

Definition 6.6 *A routing of a logical network is k-survivable if the logical network remains connected after simultaneous failures of* k *physical links.*

We introduce new notation and rewrite the necessary and sufficient conditions of the 1-survivability for the IP-over-WDM network. Then, based on the new notation, we discuss the necessary and sufficient condition for the survivable IP-over-WDM network with multiple failures.

We let $M(i,j)$ be the logical link (s,t) set whose lightpath routes through (i,j), i.e.,

$$M(i,j) = \{(s,t):\ (s,t) \text{ routes through}(i,j)\}\,,\ \forall (i,j) \in E_P. \tag{6.61}$$

The necessary and sufficient conditions in Theorem 2.1 could be rewritten as follows.

Corollary 6.1 *A routing is 1-survivable if and only if for every cut-set $CS(S, V_L \setminus S)$ of the logical topology the following holds. For every $(i,j) \in E_P$,*

$$|M(i,j) \cap CS(S, V_L \setminus S))| < |CS(S, V_L \setminus S)|. \tag{6.62}$$

In other words, no all of the logical links in the same cutset are routed through the same physical link. We can easily extend the 1-survivable condition to $k-$survivable condition, $k \geq 2$, as follows.

Theorem 6.3 *A routing is $k-$survivable if and only if every cut-set $CS(S, V_L \setminus S)$ of the logical topology the following holds. For any β pairs of physical links $(i_1, j_1), (i_2, j_2), \ldots, (i_k, j_k) \in E_P$,*

$$\left|(\cup_{p=1}^{\gamma} M(i_p, j_p)) \cap CS(S, V_L \setminus S))\right| < |CS(S, V_L \setminus S)|\,, \quad \gamma = 1, \ldots, k. \tag{6.63}$$

Next, we present another necessary and sufficient condition for k-survivability. After any physical link set (cardinality equals to k) failure, if the logical network is connected, then, there exists a corresponding spanning tree to connect every nodes in the logical network. We let $\mathcal{T}_L$ be a spanning tree in the logical network and $\mathcal{S}_{(i_q,j_q):1\leq q\leq k}$ present logical links which route through physical links (i_q, j_q) with $1 \leq q \leq k$.

Then we obtain the following conclusion.

Theorem 6.4 *A routing is k-survivable if and only if after any k physical links* $(i_1, j_1), (i_2, j_2), \ldots, (i_k, j_k)$ *fail, there exists a spanning tree* $\mathcal{T}_L$ *and*

$$\mathcal{T}_L \cap \mathcal{S}_{\{(i_q,j_q):1\leq q\leq k\}} = \emptyset. \tag{6.64}$$

Now, we proposed mixed-integer constraints to capture the condition in Theorem 6.4. First, we discuss the relationship between the $y^{st}_{i_1j_1,i_2j_2,\ldots,i_\beta j_\beta}$ and y^{st}_{ij}. Here $y^{st}_{i_1j_1,i_2j_2,\ldots,i_\beta j_\beta}$ is a binary variable which is 1 if the logical link (s,t) is routed through one of the links $(i_1, j_1), (i_2, j_2), \ldots, (i_\beta, j_\beta)$. We use $y^{st}_{i_1j_1,i_2j_2}$ and y^{st}_{ij} as an instance to show their relationship.

$$y^{st}_{i_1j_1,i_2j_2} \geq y^{st}_{i_1j_1} \tag{6.65}$$

$$y^{st}_{i_1j_1,i_2j_2} \geq y^{st}_{i_2j_2} \tag{6.66}$$

$$y^{st}_{i_1j_1,i_2j_2} \leq y^{st}_{i_1j_1} + y^{st}_{i_2j_2} \tag{6.67}$$

Constraints (6.65) and (6.66) follow from the fact that if logical link (s,t) routes through physical link (i_1, j_1) or (i_2, j_2), then, $y^{st}_{i_1j_1,i_2j_2} = 1$; otherwise it is 0. Then, we can extend Theorem 6.3 as follows.

Proposition 6.7 *The necessary and sufficient condition for the logical network to* *remain connected after any 2 failures is that*

$$\sum_{(s,t)\in CS(S,V_L\setminus S)} y_{ij}^{st} \le |CS(S,V_L\setminus S)|-1, \qquad \forall (i,j)\in E_P, \tag{6.68}$$

$$\sum_{(s,t)\in CS(S,V_L\setminus S)} y_{i_1j_1,i_2j_2}^{st} \le |CS(S,V_L\setminus S)|-1, \qquad \forall (i_i,j_1),(i_2,j_2)\in E_P. \tag{6.69}$$

Now we extend the above to the k failure case with $k\ge 3$.

$$y_{i_1j_1,i_2j_2,\ldots,i_kj_k}^{st} \ge y_{i_1j_1}^{st} \tag{6.70}$$

$$y_{i_1j_1,i_2j_2,\ldots,i_kj_k}^{st} \ge y_{i_2j_2}^{st} \tag{6.71}$$

$$\cdots\cdots$$

$$y_{i_1j_1,i_2j_2,\ldots,i_kj_k}^{st} \ge y_{i_k,j_k}^{st} \tag{6.72}$$

$$y_{i_1j_1,i_2j_2,\ldots,i_kj_k}^{st} \le \sum_{q=1}^{k} y_{i_qj_q}^{st} \tag{6.73}$$

Constraints (6.70) to (6.72) indicate if (s,t) routes through one of $(i_1,j_1),\ldots,(i_k,j_k)$, then $y_{i_1j_1,i_2j_2,\ldots,i_\beta j_\beta}^{st}=1$. Constraint (6.73) represents that if (s,t) is not routed through any one of $(i_1,j_1),\ldots,(i_\beta,j_\beta)$, then $y_{i_1j_1,i_2j_2,\ldots,i_\beta j_\beta}^{st}=0$.

Theorem 6.5 *The necessary and sufficient condition for the logical network to remain connected after k failures is that*

$$\sum_{(s,t)\in CS(S,V_L\setminus S)} y_{ij}^{st} \le |CS(S,V_L\setminus S)|-1, \tag{6.74}$$

$$\sum_{(s,t)\in CS(S,V_L\setminus S)} y_{i_1j_1,\ldots,i_\gamma j_\gamma}^{st} \le |CS(S,V_L\setminus S)|-1, \quad \gamma=2,\ldots,k \tag{6.75}$$

where $k\ge 2$.

Next, we discuss the mixed-integer formulation based on Theorem 6.5. We introduce auxiliary variable $r_{st}^{i_1j_1,i_2,j_2,\ldots,i_qj_q}$ with $1\le q\le k$ to represent whether logical link

(s,t) routes through one or more failed k physical links or not. First, we present the relationship between r and y for the 2-failure cases.

$$r_{st}^{i_1j_1,i_2j_2} \leq 1 - (y_{i_1j_1}^{st} + y_{j_1i_1}^{st}) \tag{6.76}$$

$$r_{st}^{i_1j_1,i_2j_2} \leq 1 - (y_{i_2j_2}^{st} + y_{j_2i_2}^{st}) \tag{6.77}$$

Constraints (6.76) and (6.77) force r variable to be 0 if logical link (s,t) routes through one of logical links (i_k, j_k) with $1 \leq k \leq 2$. Constraints (6.76) and (6.77) can be directly extended to be k physical link failures case as follows:

$$r_{st}^{i_1j_1,i_2j_2,\ldots,i_kj_k} \leq 1 - (y_{i_1j_1}^{st} + y_{j_1i_1}^{st}) \tag{6.78}$$

$$r_{st}^{i_1j_1,i_2j_2,\ldots,i_kj_k} \leq 1 - (y_{i_2j_2}^{st} + y_{j_2i_2}^{st}) \tag{6.79}$$

$$\ldots\ldots$$

$$r_{st}^{i_1j_1,i_2j_2,\ldots,i_kj_k} \leq 1 - (y_{i_kj_k}^{st} + y_{j_ki_k}^{st}) \tag{6.80}$$

Based on Theorem 6.5, we have the following conclusion.

Theorem 6.6 *For $k-$survivability, the necessary and sufficient conditions is as follows:*

Multiple Failure Survivable Conditions:

$$\sum_{(s,t)\in E_L} r_{st}^{i_1j_1,\ldots,i_kj_k} - \sum_{(s,t)\in E_L} r_{ts}^{i_1j_1,\ldots,i_kj_k} = -1, \qquad s = v_0 \tag{6.81}$$

$$\sum_{(s,t)\in E_L} r_{st}^{i_1j_1,\ldots,i_kj_k} - \sum_{(s,t)\in E_L} r_{ts}^{i_1j_1,\ldots,i_kj_k} = \frac{1}{|V_L|-1}, \qquad s \neq v_0 \tag{6.82}$$

$$r_{st}^{i_1j_1,\ldots,i_kj_k} \geq 0 \tag{6.83}$$

where v_0 is a picked root node in the logical network and r satisfies constraints (6.78) *to* (6.80)*.*

The proof of the above is similar to the proof in Proposition 6.5. Using the condition in Theorem 6.6 in place of the survivability condition in (6.13) to (6.16) we get an MILP for multiple failure cases.

6.6 Conclusion

In this chapter, we studied generalized versions of the survivable logical topology routing problem in an IP-over-WDM optical network. Specifically, we defined the concepts of weakly survivable lightpath routing, and strongly survivable routing in a capacitated network. In Section 6.3 and 6.4 we studied two problems. Problem 6.1 is to determine a lightpath routing that guarantees weak survivability and maximizes the logical link demands satisfaction after a physical link failure. Problem 6.2 is to determine a lightpath routing that guarantee strong survivability under minimum spare capacity requirements. For both these problems we provided MILP formulations. These formulations provide general frameworks that can be used to accommodate other scenarios such as those involving load balancing and fair capacity allocation constraints. Since MILP formulations require excessive computational time, we described heuristics for both these problems that will be effective in the case of large scale networks. We provided a comparative evaluation of the MILP formulations and our heuristics. Practical networks are adopted as the physical topologies in our experimental design. We observed that in most cases our heuristics have provided results that are very close to the optimal solution, while consuming much less computation time and also memory space even for graphs with a few dozen nodes. Thus our heuristics are suitable and effective for studying large scale problems.

In Section 6.5 we studied several extensions of the above formulations accommodating different criteria. Specifically, we provided the following:

Augmentation

- An MILP formulation to add additional links to the original logical topology that guarantees a survivable routing.

  Load Balancing

- MILP formulations to achieve load balancing, that is, to achieve allocation of logical demands in a fair manner: absolute fairness and ratio-weighted fairness.

  MILP

- An MILP formulation for survivable routing under the multiple failure scenario.

Part of the work in this chapter has been reported in [24].

Chapter 7

Summary and Future Work

7.1 Summary

The survivable logical topology mapping problem in an IP-over-WDM network deals with the cascading effect of link failures from the physical layer to the logical layer. Multiple logical links may get disconnected due to a single physical link failure, which may cause the disconnection of the logical network. In this dissertation we studied survivability issues in IP-over-WDM networks with respect to various criteria.

We first gave an overview of the two major lines of pioneering works for the survivable design problem. Though theoretically elegant, the first approach which uses ILP formulations suffers from the drawback of scalability. The second approach, namely, the structural approach, utilizes the concept of duality between circuits and cutsets in a graph and is based on an algorithmic framework called SMART. Several SMART-based algorithms have been proposed in the literature.

In order to generate survivable routing, the SMART-based algorithms require the existence of disjoint lightpaths for certain groups of logical links in the physical topology, which might not always exist. Therefore, we proposed in Chapter 4 an approach to augment the logical topology with new logical links to guarantee survivability. We first identified a logical topology that admits a survivable mapping against one physical link failure. We then generalized these results to achieve augmentation for survivability of a given logical topology to survive multiple physical link failures. Following this, we proposed in Chapter 5 a generalized version of SMART-based algorithms and introduced the concept of robustness of an algorithm which captures the ability of the algorithm to provide survivability against multiple physical link failures. We demonstrated that even when a SMART-based algorithm cannot be guaranteed to provide survivability against multiple physical link failures, its robustness could be very high.

Most previous works on the survivable logical topology design problem in IP-over-WDM networks did not consider physical capacities and logical demands. In Chapter 6, we studied this problem taking into account logical link demands and physical link capacities. We defined weak survivability and strong survivability in capacitated IP-over-WDM networks. Two-stage MILP formulations and heuristics to solve the survivable design problems were proposed. Based on the 2-stage MILP framework, we also proposed several extensions to the weakly survivable design problem, considering several performance criteria. Noting that for some logical networks a survivable mapping may not exist, which prohibits us from applying the 2-stage MILP approach, our first extension is to augment the logical network using an MILP formulation to guarantee the existence of a survivable routing. We then proposed approaches to balance the logical demands satisfying absolute or ratio-weighted fairness. Finally we showed how to formulate the survivable logical topology design problem as an MILP for the multiple failure case.

Next we conclude with an outline of two promising new directions of research.

7.2 Future Work

In this section we give an outline of two directions of future work.

7.2.1 Benders' Decomposition for the Weakly Survivable Routing Problem

In Section 6.3.1 we presented an MILP formulation of the weakly survivable mapping problem. Though theoretically elegant, solving an MILP formulation even for networks of moderate size becomes computationally very expensive. So to alleviate this problem in mathematical programming literature multi-stage programming optimization algorithms have been reported. Benders' decomposition technique is one such approach.

One direction of future research is to study the application of Benders' decomposition to the formulation given in Section 6.3.1. Now we gave an introduction to Benders' decomposition as described in [55][56]. We then show how our formulation in Section 6.3.1 can be restated to suit the application of Benders' decomposition and point out our future direction of research.

7.2.2 Benders' Decomposition

Consider the following problem,

Problem 7.1

$$\min c^T x + f^T y \tag{7.1a}$$

$$\text{subject to: } Ax + By = b \tag{7.1b}$$

$$x \geq 0, y \in Y \subseteq \mathcal{R}^q \tag{7.1c}$$

where x and y are vectors of continuous variables having dimensions p and q, respectively, Y is a polyhedron, A, B are matrices, and b, c, f are vectors having appropriate dimensions. Suppose that y-variables are *complicating variables* in the sense that the problem becomes significantly easier to solve if y-variables are fixed, perhaps due to a special structure inherent in matrix A. Benders' decomposition partitions Problem 7.1 into two problems: a master problem that contains the y-variables, and a sub-problem that contains the x-variables. We first note that Problem 7.1 can be written in terms of the y-variables as follows:

Problem 7.2

$$\min f^T y + q(y) \tag{7.2a}$$

$$\text{subject to: } y \in Y \tag{7.2b}$$

where $q(y)$ is defined to be the optimal objective function value of

Problem 7.3

$$\min c^T x \tag{7.3a}$$

$$\text{subject to: } y \in Y \tag{7.3b}$$

$$x \geq 0. \tag{7.3c}$$

Formulation in Problem 7.3 is a linear program for any given value of $y \in Y$. Note that if Problem 7.3 is unbounded for some $y \in Y$, then Problem 7.2 is also unbounded, which in turn implies unboundedness of the original Problem 7.1. Assuming boundedness of Problem 7.3, we can also calculate $q(y)$ by solving its dual. Let us associate dual variables α with constraints (7.3b). Then, the dual of Problem 7.3 is

Problem 7.4

$$\max \alpha^T (b - By) \tag{7.4a}$$

$$\text{subject to: } A^T \alpha \leq c \tag{7.4b}$$

$$\alpha \text{ unrectricted.} \tag{7.4c}$$

A key observation is that feasible region of the dual formulation does not depend on the value of y, which only affects the objective function. Therefore, if the dual feasible region (7.4b)–(7.4c) is empty, then either the primal Problem 7.3 is unbounded for some $y \in Y$ (in which case the original Problem 7.1 is unbounded), or the primal feasible region (7.3b)–(7.3c) is also empty for all $y \in Y$ (in which case Problem 7.1 is also infeasible.) Assuming that the feasible region defined by (7.4b)–(7.4c) is not empty, we can enumerate all extreme points $(\alpha_p^1, \ldots, \alpha_p^I)$ and extreme rays $(\alpha_r^1, \ldots, \alpha_r^J)$ of the feasible region, where I and J are the numbers of extreme points and extreme rays of (7.4b)–(7.4c), respectively. Then, for a given $\hat{y}$-vector, the dual problem can be solved by checking

- whether $(\alpha_r^j)^T(b - B\hat{y}) > 0$ for an extreme ray α_r^j, in which case the dual formulation is unbounded and the primal formulation is infeasible, and
- finding an extreme point α_p^i that maximizes the value of the objective function $(\alpha_p^i)^T(b - B\hat{y})$, in which case both primal and dual formulations have finite optimal solutions.

Based on this idea, the dual Problem 7.4 can be reformulated as follows:

Problem 7.5

$$\min q \tag{7.5a}$$

$$\text{subject to: } (\alpha_r^j)^T(b - By) > 0, \qquad \forall j = 1, \ldots, J \tag{7.5b}$$

$$(\alpha_p^i)^T(b - By) \leq q, \qquad \forall i = i, \ldots, I \tag{7.5c}$$

$$q \text{ unrestricted.} \tag{7.5d}$$

Note that Problem 7.5 consists of a single variable q and, typically, a large number of constraints. Now we can replace $q(y)$ in (7.2a) with Problem 7.5 and obtain a reformulation of the original problem in terms of q and y-variables:

Problem 7.6

$$\min f^T y + q \tag{7.6a}$$

$$\text{subject to: } (\alpha_r^j)^T(b - By) \leq 0, \qquad \forall j = 1, \ldots, J \tag{7.6b}$$

$$(\alpha_p^i)^T(b - By) \leq q, \qquad \forall i = 1, \ldots, I \tag{7.6c}$$

$$y \in Y,\ q \text{ unrestricted.} \tag{7.6d}$$

Since there is typically an exponential number of extreme points and extreme rays of the dual formulation in Problem 7.4, generating all constraints of type (7.6b) and (7.6c) is not practical. Instead, Benders' decomposition starts with a subset of these constraints, and solves a *relaxed master problem*, which yields a candidate optimal solution (y^*, q^*). It then solves the dual subproblem 7.4 to calculate $q(y^*)$. If the subproblem has an optimal solution having $q(y^*) = q^*$, then the algorithm stops. Otherwise, if the dual subproblem is unbounded, then a constraint of type (7.6b) is generated and added to the relaxed master problem, which is then re-solved. (Constraints of type (7.6b) are referred to as *Benders' feasibility cuts* because they enforce necessary conditions for feasibility of the primal subproblem 7.3.) Similarly, if the subproblem has an optimal solution having $q(y^*) > q^*$, then a constraint of type (7.6c) is added to the relaxed master problem, and the relaxed master problem is re-solved. (Constraints of type (7.6c) are called *Benders' optimality cuts* because they are based on optimality conditions of the subproblem.) Since I and J are finite, and new feasibility or optimality cuts are generated in each iteration, this method converges to an optimal solution in a finite number of iterations [55].

7.2.3 Benders' Decomposition Application for Survivable Routing in IP-over-WDM Networks

In this section, we consider the Benders' decomposition for the formulations in Section 6.3.1. First, we introduce the notations for the Benders' decomposition. Given $y = y_{ij}^{st}$, let

$$\Omega(y) = \{(f, r) : f \in R_{+}^{|E_L| \times |E_P|}, r \in R_{+}^{|E_L| \times |E_P|} : (6.6) - (6.16)\} \quad (7.7)$$

$$\mathcal{F}(y) = \{f : f \in \mathcal{R}_{+}^{|E_L| \times |E_P|}, (6.6) - (6.12)\} \quad (7.8)$$

$$\Gamma(y) = \{r : r \in \mathcal{R}_{+}^{|E_L| \times |E_P|}, (6.13) - (6.16)\}. \quad (7.9)$$

It is obvious that for a given y, $\Omega = \mathcal{F}(y) \cup \Gamma(y)$ and $\mathcal{F}(y) \cap \Gamma(y) = \emptyset$ since there is no relationship between f and r in the constraints. Note here, in our problem, y is a binary variable and f and r are fractional variables. Thus, our problem is a mixed-integer problem in the form of Benders' decomposition structure.

The above formulation can be rewritten as follows:

$$\min \sum_{(i,j) \in G_P, (s,t) \in G_L} y_{ij}^{st} + q(y) \quad (7.10)$$

$$\text{subject to:} \quad \text{Constraints } (6.2) - (6.5) \quad (7.11)$$

$$y \in \mathcal{R}^{|E_P| \times |E_L|} \quad (7.12)$$

where

$$q(y) = \min \sum_{(s,t) \in E_L} \xi_{st} \quad (7.13)$$

$$\text{subject to:} \quad \text{Constraints } (6.6) - (6.16) \quad (7.14)$$

$$f \in \mathcal{R}_{+}^{|E_P| \times |E_L|}, r \in \mathcal{R}_{+}^{|E_P| \times |E_L|}, \xi \in \mathcal{R}_{+}^{|E_L|}. \quad (7.15)$$

In terms of the Benders' decomposition, we decompose our problem into master problem and subproblem. The master problem in the decomposition of our problem is as follows:

$$\min \sum_{(i,j)\in G_P,(s,t)\in G_L} y_{ij}^{st} + \eta \tag{7.16}$$

$$\text{subject to:} \quad \eta \geq u^k(b - Ay) \tag{7.17}$$

$$v^j(b - Ay) \geq 0 \tag{7.18}$$

$$\text{Constraints}\,(6.2) - (6.5) \tag{7.19}$$

where $\{u^k \in R, k \in K\}$ and $\{v^j \in R, j \in J\}$ are the extreme points and extreme ray of the following subproblem. Vector b represents the right-hand-side vector and matrix A represents the constraint matrix in front of variable y in the following subproblem.

The subproblem is as follows:

$$\min \sum_{(s,t)\in E_L} \xi_{st} \tag{7.20}$$

$$\text{subject to:} \quad f \in \mathcal{F}(y) \tag{7.21}$$

$$r \in \Gamma(y). \tag{7.22}$$

To obtain Benders cuts (7.17) and (7.18), we analyze the subproblem and obtain the extreme points and extreme rays of the subproblem. Because $\Omega(y) = \mathcal{F}(y) \cup \Gamma(y)$ and $\mathcal{F}(y) \cap \Gamma(y) = \emptyset$, the above subproblem can be decomposed as follows:

Subproblem: flow part (SFF)

$$\min \sum_{(s,t)\in E_L} \xi_{st} \tag{7.23}$$

$$\text{subject to:} \quad f \in \mathcal{F}(y) \tag{7.24}$$

Subproblem: survivable part (SSF)

$$\min 0 \tag{7.25}$$

$$\text{subject to: } r \in \Gamma(y). \tag{7.26}$$

For future research we will investigate how to generate Benders' cut based on (SFF) and (SSF), respectively.

7.3 A Novel Approach to the Survivable Logical Topology Mapping Problem

In this section we present certain preliminary ideas of a novel approach to the SLTM problem. This is also a structural approach but very different from the approach as in [27][28].

Given a physical topology $G_P = (V_P, E_P)$ and the corresponding logical topology. Let M be a mapping of the logical links into lightpaths in G_P. Let $M(e) = p^e$ where p^e is the lightpath corresponding to the logical link e. Consider now a spanning tree τ of G_L. Then we denote $M(\tau)$ as the set of all physical edges used in the lightpaths corresponding to the branches in τ.

Let $M^C(\tau) = G_P \backslash M(\tau)$. Note that after the failure of one or more physical edges in $M^C(\tau)$ the tree τ will still be in G_L and so G_L will remain connected after such failures. Spanning tree τ is called a *protection tree* for the edges in $M^C(\tau)$.

A collection S of spanning trees is called a protection tree set if $\bigcup_{\tau \in S} M^C(\tau) = G_P$. Note that the above condition implies that every physical edge is in some $M^C(\tau)$ for some $\tau \in S$. So failure of any physical edge will not disconnect G_L. In view of the above we have the following.

A mapping M of logical links into lightpaths in G_P is survivable if there exists a set

S of spanning trees such that $\bigcup_{\tau \in S} M^C(\tau) = G_P$. The survivable mapping problem is to find a protection tree set with the above property. The above formulation of the survivable logical topology mapping problem gives rise to several challenging problems:

- Design an algorithm to find a mapping and the corresponding protection tree set.
- Design an algorithm to find a mapping that leads to a protection tree set S with minimum cardinality.

If the cardinality of S is small then for each tree $\tau \in S$, $M^C(\tau)$ can be expected to be large. Such a tree will be able to protect G_L against more number of failures (such as SRLG failures). We believe this direction of research has significant potential to provide insight to the SLTM problem under multiple physical edge failures.

Bibliography

[1] J. Sakaguchi, Y. Awaji, N. Wada, A. Kanno, T. Kawanishi, T. Hayashi, T. Taru, T. Kobayashi, and M. Watanabe, "109-Tb/s (7x97x172-Gb/s SDM/WDM/PDM) QPSK transmission through 16.8-km homogeneous multi-core fiber," *Proc. of National Fiber Optic Engineers Conference*, March 2011.

[2] A. Sano, H. Masuda, T. Kobayashi, M. Fujiwara, K. Horikoshi, E. Yoshida, Y. Miyamoto, M. Matsui, M. Mizoguchi, H. Yamazaki, Y. Sakamaki, and H. Ishii, "69.1-Tb/s (432 x 171-Gb/s) C- and extended L-band transmission over 240 km using PDM-16-QAM modulation and digital coherent detection," *Proc. of Optical Fiber Communication Conference*, 2010.

[3] A. Gnauck, G. Charlet, P. Tran, P. Winzer, C. Doerr, J. Centanni, E. Burrows, T. Kawanishi, T. Sakamoto, and K. Higuma, "25.6-Tb/s WDM transmission of polarization-multiplexed RZ-DQPSK signals," *IEEE/OSA Journal of Lightwave Technology*, vol. 26, no. 1, pp. 79–84, 2008.

[4] D. Clark, "Heavy traffic drives networks to IP over SONET," *Computer*, vol. 31, pp. 17–20, 1998.

[5] E. Modiano and A. Narula-Tam, "Survivable routing of logical topologies in WDM networks," *Proc. of IEEE INFOCOM*, pp. 348–357, 2001.

[6] T. Stern and K. Bala, *Multiwavelength Optical Networks: A Layered Approach.* Boston, MA,: Addison-Wesley Longman Publishing Co., 1999.

[7] C. Glingener, "Next generation optical networks and related components," *Microsystem Technologies*, vol. 9, pp. 286–290, 2003.

[8] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: an approach to high bandwidth optical WAN's," *IEEE Transactions on Communications*, vol. 40, pp. 1171–1182, 1992.

[9] N. Ghani, S. Dixit, and T.-S. Wang, "On IP-WDM integration: a retrospective," *IEEE Communications Magazine*, vol. 41, pp. 42–45, 2003.

[10] W. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration," *IEEE International Conference on Communications*, vol. 1, pp. 537 –543, 1998.

[11] M.-S. Ryu and H.-S. Park, "Survivable network design using restricted p-cycle," *Proc. of International Conference on Information Networking*, vol. 2662, pp. 918–927, 2003.

[12] C. J. Hou, "Design of a fast restoration mechanism for virtual path-based ATM networks," *Proc. of IEEE INFOCOM*, vol. 1, pp. 361–369, 1997.

[13] K. Murakami and H. Kim, "Comparative study on restoration schemes of survivable ATM networks," *Proc. of IEEE INFOCOM*, pp. 345–352, 1997.

[14] L. Sahasrabuddhe, S. Ramamurthy, and B. Mukherjee, "Fault management in IP-over-WDM networks: WDM protection versus IP restoration," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 21–33, 2002.

[15] A. Fumagalli and L. Valcarenghi, "IP restoration vs. WDM protection: Is there an optimal choice?" *IEEE Network*, vol. 14, pp. 34–41, 2000.

[16] K. Menger, "Zur allgemeinen kurventheorie," *Fundamenta Mathematicae*, pp. 96–115, 1927.

[17] H. Whitney, "Non-separable and planar graphs," *Transactions of the American Mathematical Society*, vol. 34, pp. 339–362, 1932.

[18] E. Modiano and A. Narula-Tam, "Survivable lightpath routing: a new approach to the design of WDM-based networks," *IEEE Journal of Selected Areas in Communication*, pp. 800–809, May 2002.

[19] A. Todimala and B. Ramamurthy, "A scalable approach for survivable virtual topology routing in optical WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, pp. 63–69, 2007.

[20] A. K. Todimala and B. Ramamurthy, "Survivable virtual topology routing under shared risk link groups in WDM networks," *IEEE BROADNETS*, pp. 130–139, 2004.

[21] K. Lee and E. Modiano, "Cross-layer survivability in WDM networks with multiple failures," *Proc. of IEEE Conference on Optical Fiber Communication/National Fiber Optic Engineers*, February 2008.

[22] D. D. Kan, A. Narula-Tam, and E. Modiano, "Lightpath routing and capacity assignment for survivable IP-over-WDM networks," *Proc. of Seventh International Workshop on Design of Reliable Communication Networks*, pp. 37–44, 2009.

[23] K. Lee, H.-W. Lee, and E. Modiano, "Reliability in layered networks with random link failures," *IEEE/ACM Transactions on Networking*, 2011.

[24] T. Lin, Z. Zhou, and K. Thulasiraman, "Logical topology survivability in IP-over-WDM networks: Survivable lightpath routing for maximum logical topology capacity and minimum spare capacity requirements," *Proc. of Eighth International Workshop on Design of Reliable Communication Networks*, 2011.

[25] Q. Deng, G. Sasaki, and C.-F. Su, "Survivable IP over WDM: a mathematical programming problem formulation," *Proc. of 40^{th} Allerton Conference on Communication, Control and Computing*, October 2002.

[26] M. Kurant and P. Thiran, "On survivable routing of mesh topologies in IP-over-WDM network," *Proc. of IEEE INFOCOM*, vol. 2, pp. 1106 – 1116, 2005.

[27] K. Thulasiraman, M. Javed, and G. Xue, "Circuits/cutsets duality and a unified algorithmic framework for survivable logical topology design in IP-over-WDM optical networks," *Proc. of IEEE INFOCOM*, pp. 1026–1034, 2009.

[28] ——, "Primal meets dual: A generalized theory of logical topology survivability in IP-over-WDM optical networks," *Proc. of Second International Conference on Communication Systems and Networks*, 2010.

[29] J. L. Gross and J. Yellen, Eds., *Handbook of Graph Theory.* CRC Press, 2003.

[30] J. Kleinberg, "Approximation algorithms for disjoint paths problems," Ph.D. dissertation, MIT, Cambridge, MA, 1996.

[31] M. Kurant and P. Thiran, "Survivable routing in IP-over-WDM networks in the presence of multiple failures," 2006. [Online]. Available: http://arxiv.org/abs/cs/0604053

[32] K. Thulasiraman, M. Javed, T. Lin, and G. Xue, "Logical topology augmentation for guaranteed survivability under multiple failures in IP-over-WDM optical network," *IEEE Advanced Networks and Telecommunication Systems*, 2009.

[33] M. Javed, K. Thulasiraman, M. A. Gaines, and G. Xue, "Survivability aware routing of logical topologies: On Thiran-Kurant approach, enhancements and evaluation," *Proc. of IEEE Global Telecommunications Conference*, 2006.

[34] M. Javed, K. Thulasiraman, and G. Xue, "Lightpaths routing for single link failure survivability in IP-over-WDM networks," *Journal of Communication and Networks*, vol. 9, pp. 394–401, 2007.

[35] M. Javed, K. Thulasiraman, and G. L. Xue, "Logical topology design for IP-over-WDM networks: A hybrid approach for minimum protection capacity," *Proc. of International Conference on Computer Communication Networks*, 2008.

[36] P. Raghavan and C. Thompson, "Randomized rounding: A technique for provably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7, no. 4, pp. 365–375, 1987.

[37] K. Thulasiraman, T. Lin, M. Javed, and G. Xue, "Logical topology augmentation for guaranteed survivability under multiple failures in IP-over-WDM optical networks," *Optical Switching and Networking*, pp. 206–214, 2010.

[38] K. Thulasiraman, T. Lin, and Z. Zhou, "Robustness of logical topology mapping algorithms for survivability against multiple failures in an IP-over-WDM optical network," *Proc. of Fourth International Conference on Communication Systems and Networks*, 2012.

[39] F. Ducatelle and L. Gambardella, "Fastsurv: a new efficient local search algorithm for survivable routing in WDM networks," *Proc. of IEEE Global Telecommunications Conference*, pp. 1925–1929, 2004.

[40] O. Crochat, J. L. Boudec, and O. Gerstel, "Protection interoperability for WDM optical networks," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 384–395, 2000.

[41] K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms*. Wiley-Interscience, 1992.

[42] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[43] M. Kurant and P. Thiran, "Survivable routing of mesh topologies in IP-over-WDM networks by recursive graph contraction," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 5, pp. 922–933, 2007.

[44] D. West, *Introduction to Graph Theory*. Prentice Hall, 1996.

[45] "Library for efficient modeling and optimization in networks (LEMON)." [Online]. Available: http://lemon.cs.elte.hu/trac/lemon

[46] K. Lee and E. Modiano, "Cross-layer survivability in WDM-based networks," *Proc. of IEEE INFOCOM*, pp. 1017 – 1025, 2009.

[47] H. Kerivin, D. Nace, and T. Pham, "Design of capacitated survivable networks with a single facility," *IEEE/ACM Transactions on Networking*, vol. 13, pp. 248–261, 2005.

[48] G. Anandalingam and S. Raghavan, Eds., *Telecommunications Network Design and Management*. Kluwer Academic Publishers, 2003.

[49] D. Rajan and A. Atamtürk, "A directed cycle-based column-and-cut generation method for capacitated survivable network design," *Networks*, vol. 43, no. 4, pp. 201–211, 2004.

[50] Y. Liu, D. Tipper, and K. Vajanapoom, "Spare capacity allocation in two-layer networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, pp. 974–986, 2007.

[51] X. Chu, H. Yin, and X.-Y. Li, "Lightpath rerouting in wavelength-routed WDM networks," *Journal of Optical Networking*, vol. 7, no. 8, pp. 721–735, August 2008.

[52] H. Zhang and A. Durresi, "Differentiated multi-layer survivability in IP/WDM networks," *Proc. of IEEE/IFIP Network Operations and Management Symposium*, pp. 681–694, 2002.

[53] K. S. Ho and K. W. Cheung, "Generalized survivable network," *IEEE/ACM Transactions on Networking*, vol. 15, pp. 750–760, August 2007.

[54] A. K. Chandra, D. S. Hirschberg, and C. K. Wong, "Approximate algorithms for some generalized knapsack problems," *Theoretical Computer Science*, vol. 3, pp. 293 – 304, 1976.

[55] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, pp. 238–252, 1962.

[56] J. J. Cochran, Ed., *Encyclopedia of Operations Research and Management Science.* Wiley, 2010.