SURVIVABLE DESIGN AND ANALYSIS OF WDM MESH NETWORKS

WEI HUO

A THESIS
IN
THE DEPARTMENT
OF
ELECTRICAL AND COMPUTER ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF APPLIED SCIENCE
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

NOVEMBER 2005

Your file Votre référence
ISBN: 0-494-14262-6
Our file Notre référence
ISBN: 0-494-14262-6

Abstract

Survivable Design and Analysis of WDM Mesh Networks

Wei Huo

Optical communication networks employing wavelength-division multiplexing (WDM) are currently being studied and commercially deployed to satisfy our increasing bandwidth requirements because, by using WDM technologies, an optical fiber can carry multiple non-overlapping wavelength channels, each of which typically operates at the transmission rate of 10 Gbps or more. In such a network, the failure of a network element, e.g., a fiber, a node, can cause the failure of multiple wavelength channels, thus leading to large data and service loss. For this reason, survivability in WDM networks has become a major concern for network service and equipment providers. Significant research and development effort, both in industry and in academic, are currently underway to address this issue.

In this thesis, we are particularly interested in studying the impact of network element failure(s) on network survivability. Namely, we propose and analyze a series of models and schemes to protect and restore the affected services in the networks, thus achieve a better survivability in optical networks. In additions to an introduction of optical networks and a survey of the related work, this thesis first focuses on the problem of fast recovery in Chapter 3. By using the framework of Offset-Time restoration, a novel model based on time-driven scheduling is proposed. It substantially shortens the restoration time and can be applied in both single-link failure and dual-link failure scenarios. Next, capacity reprovisioning, as a simple and efficient mechanism to protect a network against multiple failures, is investigated and a new reprovisioning scheme is proposed in Chapter 4. Finally, the application of capacity reprovisioning in traffic grooming is considered. Two frameworks, i.e., lightpath level reprovisioning and connection level reprovisioning, are proposed in Chapter 5 to improve the survivability of optical networks with grooming capability.

We show the feasibility of all proposed models and schemes by conducting comprehensive experiments and simulations, where performance is evaluated. The thesis aims to provide solutions to the design of control plane in optical networks that can be survivable against various failure scenarios.

Acknowledgments

First and foremost, I wish to express my sincere thanks and gratitude to my thesis supervisor, Professor Chadi M. Assi, for his expert guidance, and encouragement throughout my research work. Professor Assi's support has always been very generous and timely. His boundless enthusiasm and persistent commitment to high quality research has helped me polish every detail of my research papers. His technical advice and infinite patience were essential for the complete of this thesis. I am proud of having the opportunity to study under his supervision.

I thank my colleague, Mr. Lei Guang, with whom I have coauthored research papers. Our discussions contributed to this thesis greatly. I also thank my colleague, Mr. Ahmad Dhaini, for his technical expertise and general camaraderie. I enjoyed the cooperation with them on various projects.

I thank all the members of Concordia Institute for Information Systems Engineering (CIISE) for creating such a dynamic and collaborative environment for research and study. I thank the office staff and the systems supporting staff in CIISE for all their assistance. I also thank the Faculty of Engineering and Computer Science at Concordia University, for their superb teaching and guidance.

Last, but not least, I would like to express my warmest thanks to my wife, Molin, who has given me the best support with her endless perseverance, sacrifice and love. The thesis would not be possible without her support and encouragement. My deepest thanks go to my parents for their self-giving love and support throughout my life. Without them, I would never be able to complete my study at Concordia University.

Contents

List of Figures

List of Tables

Abbreviation

ADM	Add/Drop Multiplexer
APS	Auto Protection Switching
ATM	Asynchronous Transfer Mode
BBP	Bandwidth Blocking Probability
CD-LDP	Constraint-based Routing - Label Distribution Protocol
CLR	Connection-Level Re-provisioning
DFB	Distributed-FeedBack
DLE	Dynamic Lightpath Establishment
EDFA	Erbium-Doped Fiber Amplifier
FBM	Failure Broadcast Message
FNM	Failure Notification Message
FRM	Failure Recovery Message
GMPLS	Generalized MultiProtocol Label Switching
IETF	Internet Engineering Task Force
IP	Internet Protocol
LLR	Lightpath-Level Re-provisioning
LMP	Link Management Protocol
MPAC	Mixed Protection At Connection-level
OC	Optical Carrier
OSPF	Open Shortest Path First
OT	Offset Time

OT-UB	Offset Time based Upper Bound
PAC	Protection At Connection-level
PAC-HCL	Protection At Connection-level - Hop Count Limit
PAL	Protection At Lightpath-level
PON	Passive Optical Networks
OPSN	Optical Packet Switched Networks
OBSN	Optical Burst Switched Networks
RSVP-TE	Resource reSerVation Protocol - Traffic Engineering
RT	Restoration Time
RWA	Routing and Wavelength Assignment
SDH	Synchronous Digital Hierarchy
SI	Sharability Index
SLE	Static Lightpath Establishment
SLM	Single-Longitudinal Mode
SMF	Single-Mode Fiber
SONET	Synchronous Optical Network
SPAC	Separated Protection At Connection-level
STG	Survivable Traffic Grooming
STS	Synchronous Transport Signal
UNI	User-Network Interface
TDM	Time Division Multiplexing
OADM	Optical Add/Drop Multiplexer
WADM	Wavelength Add/Drop Multiplexer
WDM	Wavelength Division Multiplexing
WRN	Wavelength-Routed Networks
WXC	Wavelength Cross-Connection

Chapter 1

Introduction

The change in the fundamental character of backbone network traffic, as demonstrated by the current shift in the telecommunications industry from traditional voice-centric circuit-switched networking paradigm to data-centric packet-optimized networking paradigm, is leading to revolutionary changes in the traditional concepts of how networks are constructed. The primary reason for the paradigm shift in network design is the nature of the traffic crossing today's long-haul backbones. Specifically, Internet Protocol (IP) applications are the fastest growing segment of a service provider's network traffic. This growth is expected to continue well into this century. Fortunately, the Internet's birth coincided with several technological advances that have enabled carriers to cope with such explosive growth in data traffic. Evidently, the most important development is the rise of optical technology.

1.1 Evolution of Optical Networks

Fiber-optic technology can be considered our savior for meeting our above-mentioned need because of its potentially limitless capabilities [1, 2]: huge bandwidth (nearly 50 terabits per second (Tb/s)), low signal attenuation (as low as 0.2dB/km) and distortion, low power requirement, low material usage, and so on.

The need for optical standards led to the creation of the first generation of optical networks - Synchronous Optical NETwork (SONET) and Synchronous Digital Hierarchy (SDH) techniques [3] in the early 1980s. SONET and SDH are synchronous networks that use Time Division Multiplexing (TDM) across a ring or mesh physical topology. By far, the most common topology for SONET/SDH networks is the ring because the first generation optical networks were deployed before techniques to manage complex mesh networks were developed. The defining characteristic of "first generation" optical networks is that the optical signal is converted to electronic at each node (e.g., Add/Drop Multiplexer (ADM)). As the demand for bandwidth continues to increase, this form of traditional TDM comes under pressure from the electronic TDM bottleneck. Another major issue is the static and cumbersome process of provisioning end-to-end circuits. Unfortunately, SONET/SDH was never designed to deal with these problems.

SONET/SDH's inability to handle the huge influx of data traffic, and the associated unpredictable traffic patterns, signaled the end of this standard's long reign in the backbone transport network. All of this is about to change. Several dramatic advances in the optical-networking arena in recent years have emerged to challenge the traditional view of networking (e. g., routing, switching, provisioning, protection, and restoration). First, the abundance of bandwidth propelled by the explosion of WDM, for the first time, a new challenge to network architects. Whereas in the past, IP, ATM and related routing protocols have focused on managing the scarcity of bandwidth, the new challenge therefore is managing the abundance of bandwidth. Second, rapid advances in WDM technology have presented an attractive opportunity to evolve WDM technology toward an optical networking infrastructure with transport, multiplexing, switching, routing, survivability, bandwidth provisioning and performance monitoring, supported at the optical layer. On the contrary, "second generation" optical networks keep the traffic in optical form throughout the network, and convert the traffic to electronic only at the edges of the network. Furthermore, some routing, switching and grooming intelligences are moving into optical layer. When wavelength division multiplexing (WDM) came along, second generation optical networks assume WDM and the λs become the lowest level of transmission granularity [4].

Through the 1990s, the industry vigorously debated ring versus mesh-based principles for optical transport networks. Ring greatly predominated in practice, however, because rings offer the advantages of being closed transport subsystems, with unquestionably fast protection switching, and "pay as you grow" cost characteristics. At least initially people thought mesh-restoration was too complicated. One ring looks very simple, and this captivated the industry. But with time it was found that multi-ring networks are actually more complex than a single integrated mesh network. With the growing dominance of data over voice, the pendulum has swung back toward an interest in mesh networks. The primary reasons are that mesh offers greater flexibility, efficiency, and inherent support for multiple service classes [5]. The greater capacity efficiency comes from the more direct routing of working paths, the need for less spare capacity for restoration, and the avoidance of "stranded capacity" effects in rings (where one or more ring spans may exhaust while other spans of the ring have valuable but unusable remaining working capacity). Mesh-based networks also offer the prospect of fully self-organizing operation and fully automated path provisioning. Obviously, mesh-based networks are more future-proof than a corresponding set of rings. Over the past few years, more and more research work in literature have focused on mesh-based networks, which is also the assumption of my thesis.

1.2 Wavelength Division Multiplexing (WDM)

Wavelength-division multiplexing (WDM) has emerged as the most promising technology for taking the full advantage of the bandwidth potential of fiber and thereby satisfying the increasing demands for bandwidth. WDM is an approach that allows multiple WDM channels from different end-users to be multiplexed on the same fiber. Each of these channels operates at moderate data rates of around 10Gbps (OC-192) and it is likely to extend to 80Gbps in the near future. Thus, one can tap into the huge fiber bandwidth, with the corresponding challenges being the design and development of appropriate network architectures, protocols, and algorithms. Also, WDM devices are easier to implement since, generally, all components in a WDM device need to operate only at electronic speed. Research and development on optical WDM networks have matured considerably over the past few years. It is anticipated that the next generation of the Internet will employ WDM-based optical backbones.

The emergence of WDM can be attributed to a host of advances in optical component technologies [2]. The introduction of single-mode fibers (SMF) allowed for elimination of inter-modal dispersion effects and increased transmission distances and enabled dramatic increases in the bit rates. The single-longitudinal mode (SLM) distributed-feedback (DFB) laser is developed for high bit-rate and long distance transmission. The Allwave fiber has further expanded the usable fiber bandwidth because it does not have the 1385 nm "water-peak window" (which the conventional fiber has). The most significant advancement that revolutionized the industry is the invention of the Erbium-Doped Fiber Amplifier (EDFA), which amplifiers signals at many different wavelengths simultaneously, regardless of their modulation scheme and speed.

Next we briefly introduce the evolution of WDM optical networking.

WDM Point-to-Point System Figure 1.1 shows a four-channel point-to-point WDM solution where a WDM multiplexer (mux) combines four independent data streams, each on a unique wavelength, and sends them on a fiber; and a demultiplexer (demux) at the fibers receiving end separates out these data streams. Clearly, the capacity of the fiber link is now increased by a factor of 4. WDM mux/demux in point-to-point links with 160 channels is now available in industry [6], and this number is expected to increase.

Wavelength Add/Drop Multiplexer (WADM) Figure 1.2 shows a WADM having mux, demux and optical switches (one switch per wavelength) with add and drop ports. To let traffic (wavelengths) pass through undisturbed, the corresponding optical switch is set in the bar state (e.g., S_1 in Figure 1.2); otherwise the switch is set to cross state for adding/dropping necessary traffic (e.g., S_i). The functionality of add/drop multiplexing can be enhanced to an optical add/drop multiplexer (OADM) which may drop part of the traffic on a wavelength or it may drop a band of wavelengths.

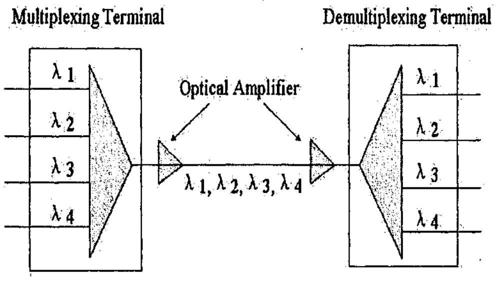


Figure 1.1: WDM Point-to-Point System

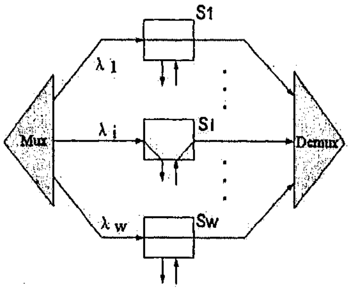


Figure 1.2: A Wavelength Add/Drop Multiplexer (WADM)

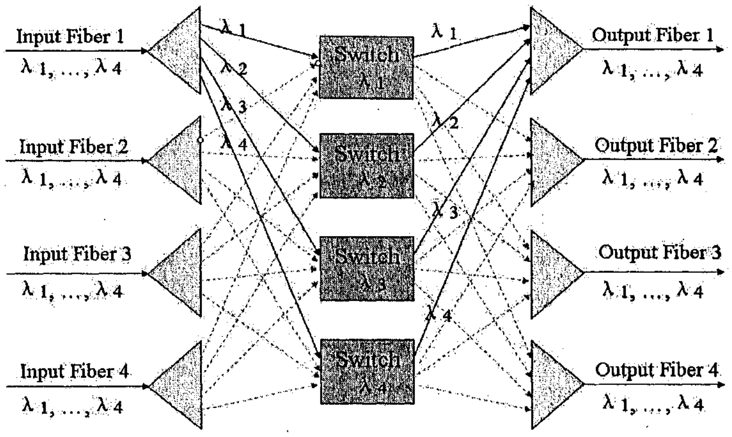


Figure 1.3: A 4 × 4 Wavelength Cross-connect (WXC)

Wavelength Cross-connect (WXC) In order to build a flexible multipoint WDM optical network, apart from WADMs, we need another optical network element called WXC. Functionally, WADM and WXC are quite similar. One main difference is that WXC can interconnect a much larger number of input fibers. Figure 1.3 shows a 4 × 4 WXC which can be realized by de-multiplexers, optical switches and multiplexers.

Todays WDM technology is evolving from a point-to-point transport technology into a networking technology. WDM-based optical networking will play a key role in the next-generation network-centric information infrastructure from local and metropolitan networks to wide-area backbones. Three categories of WDM optical networks will contribute to this vision. They are:

1. **Passive Optical Networks (PON)**: PON is also called broadcast-and-select WDM networks. Each remote node in such a network is equipped with an optical transmitter and receiver. Two fibers of opposite directions connect the node to a passive-star coupler, which broadcasts the signal arriving on each input fiber link to all outgoing fiber links. A PON only serves as the local access network, with the passive-star coupler located at the service providers central office and the remote nodes at the users premises. Currently, PON technology can support up to 128 users over each system [7].

2. **Optical Packet/Burst Switched Networks (OPSN and OBSN)**: OPSN and OBSN allow for packet-level data switching and routing at optical networking speeds. They can serve as both local and metropolitan networks. However, their practical applications and costs have to be investigated. An experimental prototype of OPSN has been developed by the European ACTS Keys to Optical Switching (KEOPS) projects [8].

3. **Wavelength-Routed Networks (WRN)**: WRN can be used for local- and metro-networks as well as wide-area backbone networks. It is a very promising candidate for the next-generation metro and wide-area networks, based on which, users can build their high-speed applications using end-to-end lightpath[1] that could be provisioned automatically and dynamically. In this thesis, I concentrate on wide-area wavelength-routed WDM networks. More background knowledge will be discussed in Chapter 2.

[1] An optical communication channel which may traverse a number of fiber links in the optical network

1.3 Motivation and Contributions

Motivated by the advance of optical technology and the desire for bandwidth with guaranteed services, we aim to improve the survivability against various failure scenarios in optical mesh networks. A network is considered to be survivable if it can maintain service continuity to the end users during the occurrence of any failures on transmission media, devices or protocols, by a suite of fault management mechanism with minimal costs, e.g., a certain period of time, some extra resources (wavelength or bandwidth). Network survivability has become a critical issue with the prevalence of WDM technology in the optical core, by which a failure of network elements (e.g., fiber, cross-connects) may cause the failure of multiple optical channels, thereby influencing a huge amount of bandwidth in transmission and leading to service interruptions to millions of users.

This thesis addresses the critical problem of how to quickly and efficiently protect/restore customer connection requests in a mesh-based wavelength-routed WDM network. It focuses on three aspects: signaling protocol for fast restoration, capacity re-provisioning and survivable traffic grooming.

In Chapter 3, we address a fundamental problem of fast recovery from network element failure(s). Current recovery protocols rely on two-way signaling schemes where upon the occurrence of the failure, the source of a failed connection initiates the recovery procedure and waits for a positive acknowledgement from the destination. We improve this scheme by proposing a new restoration mechanism called Offset-Time based restoration and we design an accurate model to estimate the Offset-Time of each failed connection using a time-driven scheduling protocol [37]. We also study the applicability of the proposed model under the scenario of double-link failures [38]. Our proposal is evaluated through simulation experiments and results show that substantial restoration gain can be achieved under various network conditions.

We study the problem of protection against multiple failures through capacity reprovisioning in Chapter 4. Capacity reprovisioning provides a mechanism by which one can find and allocate new protection capacities for the newly unprotected connections (caused by the first failure) without a priori knowledge of the location of the second failure, thus substantially improves the survivability of the network against multiple failures. We propose a new algorithm to improve the performance of re-provisioning [39, 40] and compare it with conventional schemes under both centralized and distributed implementations [41]. Furthermore, we discuss the impact of resource sharing on the performance of re-provisioning [42].

In Chapter 5, we investigate the problem of survivable traffic grooming (STG) in shared mesh WDM networks and propose different frameworks for improving the survivability of low speed demands against multiple near-simultaneous failures by capacity reprovisioning. We propose two different reprovisioning schemes (lightpath level reprovisioning, LLR, and connection level reprovisioning, CLR) and study the performance of these schemes under two grooming policies (PAL and PAC) [43, 44]. We show that while CLR reprovisions substantially more connections than LLR, CLR yields a much better network robustness to simultaneous failures due to its superior flexibility in using network resources.

1.4 Thesis Outline

The rest of the thesis is organized as follows. Chapter 2 overviews the background knowledge related with my research works in optical networks, i.e., the Routing and Wavelength Assignment (RWA) problem, the control plane issue and the mesh-based survivability. In Chapter 3, we discuss the signaling protocols in fast restoration of optical mesh networks. Capacity reprovisioning and its application in survivable traffic grooming are investigated in Chapter 4 and 5 respectively. Finally, summary and directions of future research are presented in Chapter 6.

Chapter 2

Background and Related Work

This chapter gives an overview on the state-of-the-art of optical networks. We first introduce the wavelength-routed WDM network architecture; then we discuss the Routing and Wavelength Assignment (RWA) problem and the control plane issues. Finally, we present the survivability problem in optical networks, a discussion that is fundamental of this thesis.

2.1 Architecture of Wavelength-Routed WDM Network

Wavelength-routed WDM networks is widely expected to form the basis for a future all-optical infrastructure and is built on the concept of wavelength routing. A wavelength routing network, shown in Figure 2.1, consists of optical cross-connects (OXCs) connected by a set of fiber links to form an arbitrary mesh topology. The services that a wavelength routed network offers are in the form of logical connections implemented using lightpaths. A lightpath is an optical communication channel which may traverse a number of fiber links in the optical network. Information transmitted on a lightpath does not undergo any

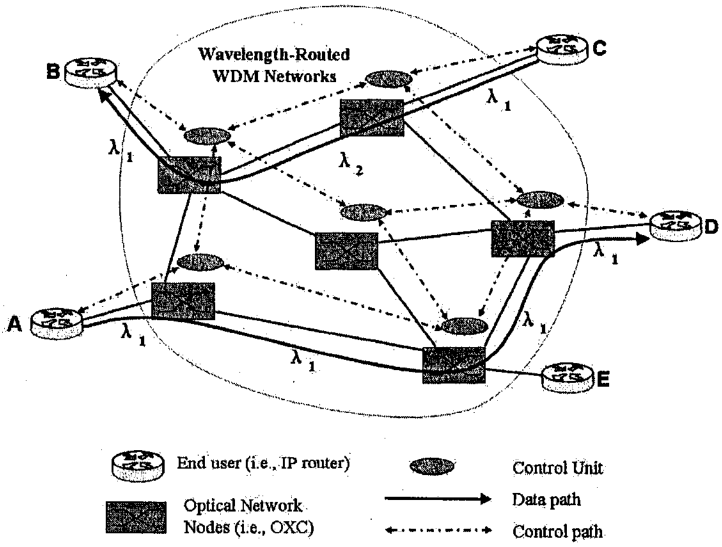


Figure 2.1: Architecture of Wavelength-Routed WDM Network

conversion to and from electrical form within the optical network, and thus, the architecture of the OXCs can be very simple because they do not need to do any signal processing.

The OXCs provide the switching and routing functions for supporting the logical data connections between client sub-networks. An OXC takes in an optical signal at each of the wavelengths at an input port, and can switch it to a particular output port, independent of the other wavelengths. An OXC with *N* input and *N* output ports capable of handling *W* wavelengths per port can be thought of as *W* independent $N \times N$ optical switches. These switches have to be preceded by a wavelength demultiplexer and followed by a wavelength multiplexer. Thus, an OXC can cross-connect the different wavelengths from the input to the output, where the connection pattern of each wavelength is independent of the others. By appropriately configuring the OXCs along the physical path, logical connections (lightpaths) may be established between any pair of sub-networks.

As Figure 2.1 illustrates, each OXC has an associated electronic control unit which is responsible for control and management functions related to setting up and tearing down lightpaths; these functions will be discussed in detail in Section 2.3. In particular, the control unit communicates directly with its OXC, and is responsible for issuing configuration commands to the OXC in order to implement a desired set of lightpath connections. The control unit also communicates with the control units of adjacent OXCs or with attached client sub-networks as shown in Figure 2.1. These lightpaths are typically implemented over administratively configured ports at each OXC and use a separate control wavelength at each fiber. Thus, we distinguish between the paths that data and control signals take in the optical network: data lightpaths originate and terminate at client sub-networks and transparently traverse the OXCs, while control lightpaths are electronically terminated at the control unit of each OXC. Communication on the control lightpaths uses a standard signaling protocol (e.g., GMPLS [9]), as well as other standard protocols necessary for carrying out important network functions. Standardization efforts, which are crucial to the seamless integration of multi-vendor optical network technology, will also be mentioned in Section 2.3.

2.2 Routing and Wavelength Assignement Problem

A unique feature of optical WDM networks is the tight coupling between routing and wavelength selection. A lightpath is implemented by selecting a path of physical links between the source and destination edge nodes, and reserving a particular wavelength on each of these links for the lightpath. Thus, in establishing an optical connection we must deal with both routing (selecting a suitable path) and wavelength assignment (allocating an available wavelength for the connection).

Given a set of connections, the problem of setting up lightpaths by routing and assigning a wavelength to each connection is called the Routing and Wavelength Assignment (RWA) problem, and is significantly more difficult than the routing problem in electronic networks. The additional complexity arises from the fact that routing and wavelength assignment are subject to the following two constraints:

1. *Wavelength continuity constraint*: a lightpath must use the same wavelength on all the links along its path from source to destination edge node.

2. *Distinct wavelength constraint*: all lightpaths using the same link (fiber) must be allocated distinct wavelengths.

The wavelength continuity constraint may be relaxed if the OXCs are equipped with wavelength converters [10]. A wavelength converter is a single input/output device that converts the wavelength of an optical signal arriving at its input port to a different wavelength as the signal departs from its output port, but otherwise leaves the optical signal unchanged. That is, wavelength conversion allows a lightpath to use different wavelengths along different physical links.

RWA problem is the fundamental control problem in optical WDM networks. Since the performance of a network depends not only on its physical resources (e.g., OXCs, converters, fibers links, number of wavelengths per fiber, etc.) but also on how it is controlled, the objective of an RWA algorithm is to achieve the best possible performance within the limits of physical constraints. The RWA problem can be cast in numerous forms. The different variants of the problem, however, can be classified under one of two broad versions [11, 12]: a static RWA, whereby the traffic requirements are known in advance, and a dynamic RWA, in which a sequence of lightpath requests arrive in some random fashion.

The static RWA problem arises naturally in the design and capacity planning phase of architecting an optical network. With static traffic, the entire set of connections is known in advance, and the problem is then to set up lightpaths for these connections in a global fashion while minimizing network resources (e.g., physical links and wavelengths). Alternatively, one may attempt to set up as many of these connections as possible for a given fixed number of wavelengths. Because off-line algorithms have knowledge of the entire set of demands (as opposed to on-line algorithms that have no knowledge of future demands), they make more efficient use of network resources and project a lower overall capacity requirement. The RWA problem for static traffic is known as the Static Lightpath Establishment (SLE) problem.

The dynamic RWA problem is encountered during the real-time network operation phase and involves the dynamic provisioning of lightpaths [35]. For the case of dynamic traffic, a lightpath is set up for each connection request as it arrives, and the lightpath is released after some finite amount of time. The objective in the dynamic traffic cases is to set up lightpaths and assign wavelengths in a manner which minimizes the amount of connection blocking, or which maximizes the number of connections that are established in the network at any time. This problem is referred to as the Dynamic Lightpath Establishment (DLE) problem.

The SLE problem can be formulated as a mixed-integer linear program [33], which is NP-complete [34]. To make the problem more tractable, SLE can be partitioned into two sub-problems: (1) routing and (2) wavelength assignment; and each sub-problem can be solved separately. The DLE problem is more difficult to solve, and therefore, heuristics methods are generally employed in these two sub-problems. Now in literature, a number of approaches and heuristics have been proposed for the routing sub-problem and wavelength assignment sub-problem. Interested readers can refer to [11] for detailed information.

2.3 Control and Management Protocols

So far we have focused on the network architecture and RWA problems with a view to achieve specific performance objectives, e.g., efficient utilization of network resources, minimize the blocking probability of the network, etc. Equally important to an operational network are associated control plane issues involved in automating the process of lightpath establishment/release and in supporting the network design and traffic engineering functions.

The vision of a future optical network which is capable of providing a bandwidth-on-demand service by dynamically creating and tearing down lightpaths between client sub-networks. There are two broad issues that need to be addressed before such a vision is realized [13]. First, a signaling mechanism is required at the user-network interface (UNI) between the client sub-networks and the optical network control plane. The signal channel allows edge nodes to dynamically request bandwidth from the optical network, and supports important functions including provisioning, neighbor and service discovery, address registration, etc. Both the ODSI [14] coalition and the OIF [15] have developed specifications for the UNI; the OIF specifications are based on GMPLS.

Second, a set of signaling and control protocols must be defined within the optical

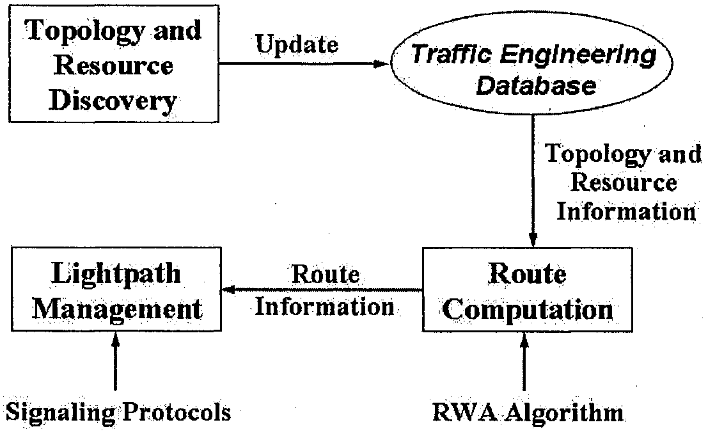

Topology and Resource Discovery
Update
Traffic Engineering Database
Topology and Resource Information
Lightpath Management
Route Information
Route Computation
Signaling Protocols
RWA Algorithm

Figure 2.2: Components of Control and Management Plane

network to support dynamic lightpath establishment/release and traffic engineering functionality. These protocols are implemented at the control module of each OXC. Currently, most of standardization activities addressing the control plane aspects of optical networks are underway [16, 35] within the Internet Engineering Task Force (IETF), reflecting a convergence of the optical networking and the IP communities to developing technology built around a single common framework, namely, GMPLS, for controlling both IP and optical network elements [18]. There are three components of the control plane that are crucial to setting up lightpaths within the optical network (refer to Figure 2.2):

1. **Topology and resource discovery.** The main purpose of discovery mechanism is to disseminate network state information including resource usage, network connectivity, link capacity availability, and special constraints.

2. **Route Computation.** This component employs RWA algorithms and traffic engineering functions to select an appropriate route for a requested lightpath.

3. **Lightpath Management.** Lightpath management is concerned with setup and teardown of lightpaths, as well as coordination of protection switching in case of failures.

Topology and resource discovery includes neighbor discovery, link monitoring and state distribution. The Link Management Protocol (LMP) [19] has been proposed to perform neighbor discovery and link monitoring. Distribution of state information is typically carried out through link state routing protocols such as OSPF [20]. In particular, the link state information that these protocols carry must be augmented to include optical resource information including: wavelength availability and bandwidth, physical layer constraints, and link protection information, among others. This information is then used to build and update the optical network traffic engineering database (see Figure 2.2) which guides the route selection algorithm.

Once a lightpath is selected, a signaling protocol must be invoked to set up and manage the connection. Two protocols have currently been defined to signal a lightpath setup: RSVP-TE [21] and CR-LDP [22]. RSVP-TE is based on the resource reservation protocol (RSVP) [23] with appropriate extensions to support traffic engineering, while CR-LDP is an extension of the label distribution protocol (LDP) [24] augmented to handle constraint-based routing. The protocols are currently being extended to support GMPLS [9, 25].

The control mechanism can be implemented using a separate channel, namely, a control channel in the WDM network. Alternately, the control information can be carried in-band, as in Multi-Protocol Label Switching (MPLS), whose wavelength-routing version is referred to as Multi-Protocol Lambda Switching (MPλS). Besides supporting the signaling protocol and the network-topology and status update protocol, the control channel should also have the ability to discover and recover from faults, which we describe below.

2.4 Network Survivability

In optical network, one single strand of fiber can provide tremendous bandwidth (a few tens of terabits per second) by multiplexing many non-overlapping wavelength channels. A fiber cut usually occurs due to a duct[1] cut during construction and destructive natural events, such as earthquakes, etc. All the lightpaths that traverse the failed fiber will be disrupted thus lead to a severe traffic loss and service interruption. With the frequent occurrence of fiber cuts and the tremendous traffic loss a failure may cause, network survivability becomes a critical concern in network design and real-time operation. Although higher protocol layers (such as ATM and IP) operating over the optical network have their own recovery procedures to recover from link failures, the recovery time is still significantly large (on the order of seconds), whereas we expect that restoration times at the optical layer will be on the order of milliseconds to minimize data losses [27]. Furthermore, it is beneficial to consider restoration mechanisms in the optical layer for the following reasons [28]: (a) the optical layer can efficiently multiplex protection resources (such as wavelengths and fibers) among several higher-layer network applications, and (b) survivability at the optical layer provides protection to higher-layer protocols that may not have built-in protection.

As networks migrate from ring to mesh, designing and operating a survivable WDM mesh network have received increasing attentions. To survive single-link failures, two basic approaches based on protection and restoration have been used and studied in the past. If backup resources (routes and wavelengths) are pre-computed and reserved in advance, we call it a protection scheme. Otherwise, when a failure occurs, if another route and a free

[1] A duct is a bidirectional physical pipe between two nodes. In practice, fibers are put into cables, which are buried into ducts under the ground.

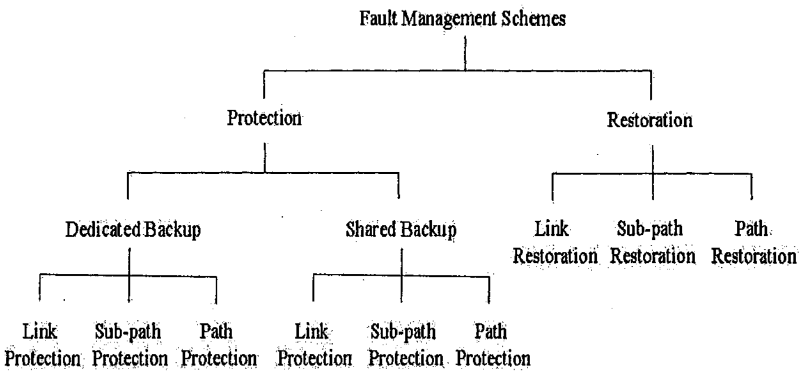


Figure 2.3: Fault Management Schemes

wavelength have to be discovered dynamically for each interrupted connection, we call it restoration scheme. Protection/restoration schemes to ensure optical network survivability can be broadly classified based on whether the resources are dedicated for protection or dynamically allocated for restoration [29, 30] (Figure 2.3). In dedicated-resource protection (which includes automatic protection switching (APS) and self-healing rings), the network resources may be dedicated for each failure scenario, or the network resources may be shared among different failure scenarios. In dynamic restoration, the spare capacity available within the network is utilized for restoring services affected by a failure. Generally, dynamic restoration schemes are more efficient in utilizing capacity due to the multiplexing of the spare-capacity requirements and provide resilience against different kinds of failures, while dedicated-resource protection schemes have a faster restoration time and provide guarantees on the restoration ability.

Protection schemes can be further classified into three categories: path, sub-path and link protection (Figure 2.4).

1. *Path protection*: In path protection, the source and destination nodes of each connection statically reserve backup paths and wavelengths during call setup. All the connections that traverse the failed link are rerouted to the backup route. The working and backup paths for a connection must be physically link-disjoint (Figure 2.4(a)).

2. *Link protection*: In link protection, all the affected connections that traverse the failed link are re-routed around that failed link. The source and destination nodes of the connections are oblivious to the link failure and re-routing (Figure 2.4(b)).

3. *Sub-path protection*: Sub-path protection has recently been studied in [31, 32] and it is achieved by dividing a primary path into a sequence of segments and protecting each segment separately (or dividing the whole network into different domain where a lightpath segment in one domain must be protected by the resource in the same domain) (Figure 2.4(c)).

Clearly, while path protection leads to efficient utilization of backup resources and lower end-to-end propagation delay for the recovered route, link protection provides shorter protection switching time. As a compromise, sub-path protection can achieve high scalability and fast recovery time for a modest sacrifice in resource utilization. Figure 2.4 shows the classification of protection and restoration schemes.

Dynamic restoration can also be classified as path, sub-path or link based depending on the type of rerouting. In path restoration, upon a link failure, the end nodes of each connection that traverses the failed link are informed about the failure. Then the end nodes independently discover a backup route on the end-to-end basis. In link restoration, the end nodes of the failed link dynamically discover a route around the failed link for each connection that traverses the link. In sub-path restoration, each disrupted connection is restored

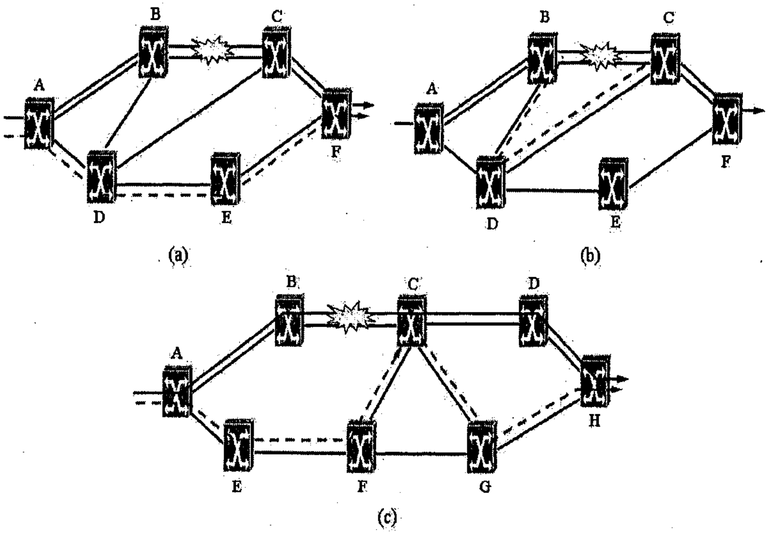


Figure 2.4: Protection Schemes

by finding a route only for the failed segment, and other segment(s) are not aware of the restoration. Among the three schemes, path restoration is the slowest and link restoration is the fastest.

Chapter 3

Improving Signaling Recovery in Shared Mesh WDM Networks

3.1 Introduction

Traditionally, ring based SONET networks offered 50ms restoration time (RT) using pre-allocated protection capacity along pre-planned protection paths [5]. Recently, mesh based networks [1, 5] have received much attention due to the increased flexibility they provide. Now, one of the key benefits of optical mesh networks is the improved bandwidth utilization due to the sharing of restoration resources across multiple failure-independent connections [29, 30, 45, 46, 47]. However, unlike automatic protection switching (APS) of SONET rings and dedicated protection of mesh networks where rapid network recovery upon failures is achieved, shared restoration exhibits increased recovery latencies [45, 47]. Shared network restoration typically involves a common set of steps, including (1) backup path selection, (2) failure detection, (3) notification, and (4) signaling recovery protocols.

Upon the detection of a failure and the receipt of a failure notification, the nodes responsible for initiating the recovery commence a signaling procedure to configure appropriate protection resources (e.g., wavelength and XC switches) for each of the failed connections [48, 36, 49]. Therefore, efficient network restoration schemes are required to eliminate or minimize the associated switching and signaling latencies.

Currently, a commonly used signaling mechanism is a two-phase process [48, 50] (also called Round-Trip restoration). When the source node of a failed connection is notified of a failure, it sends a recovery message towards the destination along its designated backup route to configure the associated protection resources. The destination upon receiving this message will prepare an acknowledgement (ACK) and forward back to notify the source of the successful setup of the backup route and finally the source node resumes its transmission. Obviously, the Round-Trip propagation delays of these recovery messages will have a significant adverse impact on the network restoration time. Moreover when a fiber link is cut, a large number of connections may fail and their recovery is initiated simultaneously; therefore a surge of restoration messages may arrive at a node, each requesting the configuration of a particular cross-connect switch. The node receiving such requests, after processing these messages, will issue the appropriate commands to configure its switch fabric. Depending on the switch architecture, these commands may be handled either sequentially, in patch or parallel [49]. Clearly, significant queuing (processing) delays of recovery messages and switching waiting times along with message propagation delays may lead to longer recovery times.

The authors of [51] highlighted this problem and proposed restoration message aggregation to reduce the impact of message queuing delays and they also showed that a switch fabric with parallel command execution could substantially cope with the switching delays. It is important, however, to note that message aggregation may have some limitations due to the constraints imposed in aggregating messages. Moreover, the network parameters that severely impact the network recovery times the most are the restoration message propagation delays and the switch configuration waiting times when deploying switches with sequential configuration, which in essence message aggregation is not meant to resolve.

Since propagation delays are expected to have larger impact on service restoration, they could yield substantial gain in achieving fast network recovery if eliminated. Therefore, we propose to modify the two-phase signaling to eliminate the impact of the message propagation delays. Upon notification, a source node starts its recovery procedure and subsequently schedules (after some offset time, OT) the transmission of its data. Unlike the offset time proposed in [36], where only upper bound is derived for each connection, here we propose a time-driven scheduling procedure to accurately estimate the offset time of each failed connection. We demonstrate that the new restoration framework presents considerable improvement over its predecessor under varying network setting parameters and regardless of the switch fabric architecture. This chapter is organized as follows. Section 3.2 gives an overview of Offset Time based Upper Bound (OT-UB) restoration. Subsequently in section 3.3 we introduce the mechanism of proposed scheduling restoration scheme. In section 3.4 we discuss the applicability of the proposed approach to situations with dual failures. Section 3.5 and 3.6 are performance evaluation and conclusion.

3.2 Offset-Time based Upper Bound Restoration

In shared mesh networks, restoration resources are reserved at the time a connection is provisioned and they are configured upon the occurrence of a failure along the path of the connection. Because these resources are already reserved, contention between two (or more) connections attempting to restore is unlikely to occur. Therefore, unlike the Round-Trip restoration, the source does not need to wait for the receiver to acknowledge the successful configuration of the backup route. Instead, the source can compute the time it takes for the restoration path to be ready upon commencing the recovery process and offsets its transmission accordingly. This offset time must be selected such that any intermediate switch along the backup path must have its cross-connect configured prior to the arrival of data. Note that when a failure occurs, a large number of connections may fail; as the restoration procedure initiates, a large number of recovery messages may simultaneously arrive at some node(s) to configure protection resources. Therefore, a message may experience major queuing delays before it is processed. To successfully select the offset times, we proposed in [36] that the source node of each affected connection be informed by the total number of failed connections (N) and accordingly it assumes a worst-case delay in computing its OT (i.e., the message is always the last to be processed at each node along the restoration path, shown in Figure 3.1). This will result in upper bound for the offset time (OT-UB):

$$T_{OT}^{1} = \sum_{i=0}^{n-1} (N \times T_P) + T_P + N \times T_{SC}; \qquad (1)$$

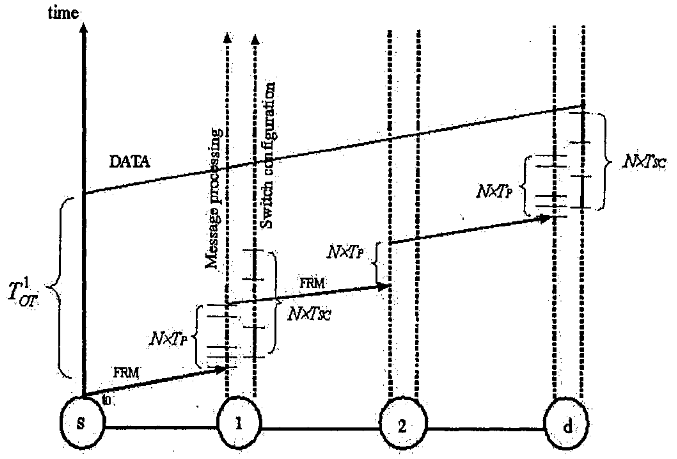


Figure 3.1: Illustrative example - Offset Time based Upper Bound Restoration

where n is the number of hops between source and destination, T_P is the message processing time, T_{SC} is the switch configuration time. Here, $\sum_{i=0}^{n-1}(N \times T_P)$ represents the upper bound queuing delays a recovery message will experience along all nodes, except the destination, on the restoration path and $T_P + N \times T_{SC}$ represents the message processing time at the destination plus the upper bound waiting time (i.e., worst case) for the particular switch to be configured. For further details, please refer to [36]. The RT therefore becomes

$$RT = T_D + T_N + T^1_{OT}; \tag{2}$$

where T_D is the failure detection time, and T_N is the failure notification time. Clearly, this approach eliminates the impact of the message propagation delays from the overall RT. However, due to the fact that no exact information is available on the size of the surge of restoration messages arriving simultaneously at a node (herein, such a node is termed as a *conflict node*) other than N, the RT could become increasingly large as the network condition changes (e.g., increasing number of wavelengths or deploying switches with longer

T_{SC} [48]) since

$$\frac{\partial T_{OT}^{1}}{T_{SC}} \propto N; \tag{3}$$

this eventually could deteriorate the performance of OT-UB scheme.

3.3 A Scheduling Approach for Rapid Restoration

Due to the fact that no exact information is available on the size of the surge of restoration messages arriving at a node and competing for configuring their protection resources other than the total number of failed connections (N), larger offset times will be selected by each node; this will result in additional unnecessary delays that could significantly increase the network RT. Also note that some of the restoration messages sharing a conflict node may not necessarily be in conflict, depending on the inter-arrival time(s) of their arrivals at the conflict node (as explained later).

Therefore, we need some mechanism that can accurately model the sequence of arrival of restoration messages at conflicting nodes and decide upon that how much delay each message experiences before its switch is configured. In other words, if we know how many recovery messages (and the order of their arrivals) may arrive simultaneously at any node along the backup route of a failed connection, then we can estimate the switching delays a connection may experience before its protection resources are configured. Accordingly, one can estimate a more accurate offset time than the upper bound. If there are more than one conflicting nodes for a failed connection, then the largest waiting time a message experiences at a conflicting node is used to compute the OT for the corresponding failed connection. Hence, the problem is reduced into a *scheduling problem* where the inputs are

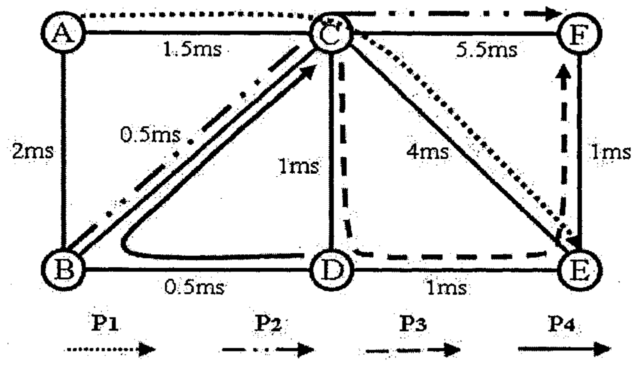


Figure 3.2: Illustrative example - network topology

(1) the network topology; (2) the set of failed connections along with (3)their restoration routes; the output will be a *timetable* where each failed connection is associated with its OT such that no conflict can occur when they start the recovery. This algorithm is "centralized" and will be executed by the node detecting the failure (i.e., upstream node of a failed link). We illustrate our approach through an example in Figure 3.1. It shows a sample network with 4 backup connections (P_1-P_4) that need to be configured upon the failure of a link f (that is not shown in the figure). The numbers on the links represent the link propagation delays in milliseconds.

Given the delays on each link in the network, the upstream node of the failed link f, Ω_f, can virtually trace the propagation of each recovery message for each failed connection going through the failed link and determine the sequence of arrival of recovery messages at each intermediate node as well as the likelihood of conflict a message may have with other recovery messages for other connections. One requirement, however, is that Ω_f must maintain a knowledge of the protection routes of all connections routing through link f; this information is made available to the node during the provisioning phase. We note here that maintaining this information does not pose any scalability problem since only information

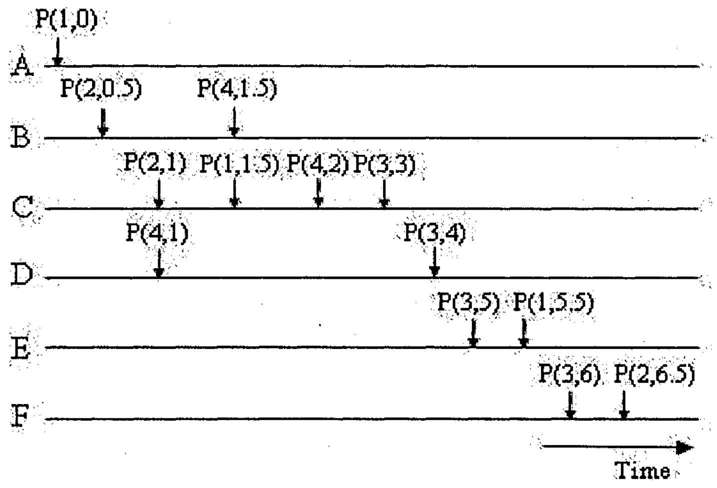


Figure 3.3: Illustrative example - sequence of messages

about connections routed through this node are to be maintained and not all connections currently in the network. The amount of this information can at most be $O(W \cdot R)$ where W is the number of wavelengths per link and R is the nodal degree (number of edges leaving this node) of this current node.

Figure 3.2 shows a time sequence of the flow of the configuration messages, as determined by Ω_f, where $P(i,j)$ represents the arrival of a recovery message for connection i at time j. We assume in this example that the notification times T_N of P_1-P_4 are 0ms, 0.5ms, 3ms, and 1ms respectively. Upon following the flow of these messages, it becomes easy to identify with the other recovery messages that a particular recovery message (of a particular failed connection) is contending with. For example, the recovery message for P_1 is the first message arriving at node A, and it is the second message arriving at nodes C and E. Therefore under worst-case assumption, it will at most be the second message to be processed along any node on its protection route and there will at most be one message waiting ahead before it is processed. Hence, its offset time parameter should be 2 whereas using the upper bound [36] scheme, the offset time is computed based on N = 4 (i.e., 4 failed connections).

We now formally explain the scheduling procedure, but first we make the following observations. The main delays we are trying to minimize in this framework are the propagation delays and the switching times as the message processing delays are considered smaller in comparison with the switching times ($T_P \ll T_{SC}$). We start by giving some variable definitions:

N : Total number of working connections going through f;

W_f=$\{w_1, w_2, \ldots, w_N\}$: The set of failed working connections going through failed link f;

s_{w_i}: Source node of connection w_i;

$T_N^{w_i}$: Time it takes to notify the source of connection;

L_{w_i}: The offset-time parameter of connection;

$L_{w_i}^j$: The offset-time parameter of connection w_i estimated at node j along its backup path;

$t_{w_i}^j$: Arrival time of recovery message for connection w_i at node j;

$\Delta(j-1,j)$: Propagation delay between node j and its upstream node. So, $t_{w_i}^j = t_{w_i}^{j-1} + N \times T_P + \Delta(j-1,j)$;

T_d: Failure detection time;

$C_{id}^{w_i}$: Unique identification for connection w_i;

Φ_j: Timetable of node j, which is initially empty. Each item of the timetable has two elements: $\{C_{id}^{w_i}, t_{w_i}^j\}$;

n_{w_i}: The number of nodes along the restoration route for connection;

T_{OT,w_i}^2: Offset time for connection w_i.

The computational complexity of Algorithm 1 is $O(L \cdot N^2 \cdot \log N)$, where N is total

Algorithm 1 Pseudo code of the Scheduling Procedure

```
for each failed connection $w_i, (i = 1, \ldots, N)$ do
    Compute its failure notification time $T_N^{w_i}$;
    for each node $j$ along the backup path for $w_i$ do
        Compute $t_{w_i}^{j} = t_{w_i}^{j-1} + N \times T_P + \Delta(j-1, j)$;
        Insert $\{C_{id}^{w_i}, t_{w_i}^{j}\}$ into $\Phi_j$;
        Sort $\Phi_j$ based on $t_{w_i}^{j}$;
        Compute $L_{w_i}^{j}$ (which is also the position of $t_{w_i}^{j}$ in $\Phi_j$)
    end for
end for
for each $w_i$ do
    Compute $L_{w_i} = \max_{k=1,2,\ldots n_{w_i}}\{L_{w_i}^{k}\}$
    Compute $T_{OT,w_i}^{2} = (n_{w_i} - 1) \times L_{w_i} \times T_P + T_P + L_{w_i} \times T_{SC}$
end for
```

number of working connections going through failed link f, L is the average length of backup paths.

Upon the failure of link f, Ω_f will run Algorithm 1 to evaluate the OT for each failed connection. Subsequently, a notification message is sent to the source node of each connection along with the computed OT to start its recovery procedure and accordingly it schedules (offsets) the restoration of its data.

Note that, here the notification time ($T_N^{w_i}$) for a connection w_i is computed as follows:

$$T_N^{w_i} = \sum_{j=\Omega_f}^{j-1=s_{w_i}} \Delta(j-1, j) + N \times T_P \times m_{w_i}; \tag{4}$$

Where the first part in the equation accounts for the propagation delays of the notification message between Ω_f and s_{w_i}, and the second part of the expression represents the total processing time of a message at a node along the notification path times the total number of hops along the notification path, m_{w_i}. Here, worst case processing delays ($N \times T_P$) are considered for a notification message as it propagates along the notification path; the reason is that notification messages for different failed connections may also contend for processing along notification route. Since T_P is typically small (few μseconds), this upper bound processing delays will not affect the performance of the recovery protocol. Finally, since the computed notification time is an upper bound expression that is used in the OT computation, it will be slightly larger than the actual notification time. Therefore, the source node upon receiving the notification message (at $T_{N,actural}^{w_i}$) it will wait ($T_N^{w_i} - T_{N,actural}^{w_i}$) before it commences its recovery, in order to avoid causing any perturbation to the computed offset times.

We note here that this scheduling algorithm provides a good approximation on the waiting time of XC switch request by estimating the number of requests waiting ahead of a recovery message. However, it ignores the inter-arrival time between two consecutive requests. After identifying the sequence of arrivals at some node, one can further check whether the inter-arrival time of two consecutive messages is larger than the T_{SC}. If so, the switch of the second request can be directly configured without any delay, since the configuration of the previous request has already terminated. This means that some messages may arrive consecutively at a conflicting node, but actually they may or may not be in conflict depending on their inter-arrival time. Otherwise, the second message will need to wait for some period of time until all requests waiting ahead are processed. As before, all conflicting nodes need to be considered, and to determine the offset time we choose the node with maximal waiting time for the request to be processed. As an example, let $\delta 1$ and $\delta 2$ be the inter-arrival time of recovery messages between (P_1, P_2) and (P_1, P_3) at nodes C and E respectively (Figure 3.2). If $(\delta 1 < T_{SC}) \bigwedge (\delta 2 < T_{SC})$, then the recovery message for connection P1 does not contend with any of the other messages and therefore smaller OT can be achieved. Obviously, this optimized approach will further complicate the scheduling procedure, so we do not discuss it any further here but we present its evaluation in section 3.5.

3.4 Double Link Failure Recovery

As single link failures are common failure scenarios, normally recovering from these failures is completed within few milliseconds to few seconds. However the time it takes to repair a link may be few hours to few days [1] and it is likely that a second failure occurs during this period, causing two links to be down near-simultaneously. Previous research has addressed the problem of routing connections under dual failure assumptions [12, 13] where extra protection capacity is preplanned in the network. In this section we are only interested in the applicability of the proposed restoration framework under double failure assumptions. We assume, as in [12], each connection is protected by two link-disjoint backup (primary and secondary) paths. We further assume that both failures occur near-simultaneously, that is the second occurs while the network is recovering from the first failure; otherwise, recovery from both failures is treated independently; i.e., similar to single link failure.

We let f_1 and f_2 be the first and second failure respectively and G_1 and G_2 be the two groups of working connections to be restored upon the failures of f_1 and f_2. Upon detecting a failure, the upstream node (Ω_{f_k}) of the failed link ($f_k, k = 1,2$) will send a notification message ($FNM(f_k)$) to the source node of each failed connection and simultaneously it broadcasts a failure message ($FBM(f_k)$) to all nodes in the network. We assume two links

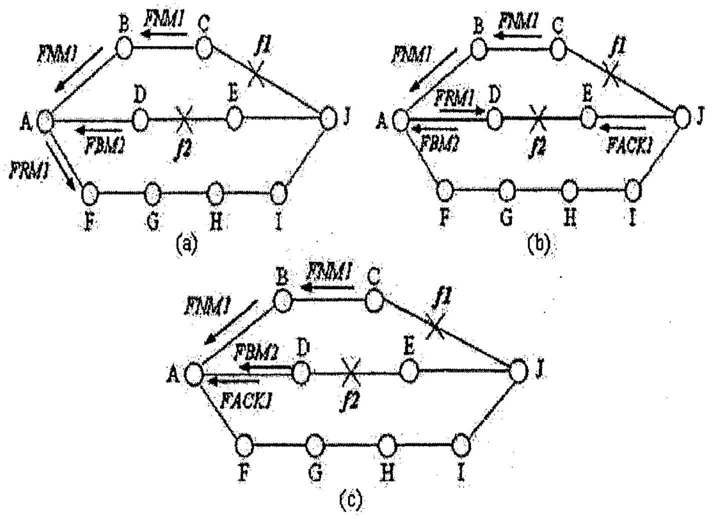


Figure 3.4: Illustrative example - Signaling in Dual-Link Failure

can fail near-simultaneously in any arbitrary order. We also define: $b_1^{i,1}$, and $b_1^{i,2}$: the primary and secondary backup paths respectively for a connection $w_i^{G_1}$ in G_1. $b_2^{i,1}$, and $b_2^{i,2}$: the primary and secondary backup paths respectively for a connection $w_i^{G_2}$ in G_2.

3.4.1 Conventional Recovery Procedure, Case I

We start first by studying the conventional (Round-Trip) restoration scheme under dual failures. There are two possible scenarios to be considered:

A If the second failure (f_2) does not impact the first primary backup ($b_1^{i,1}$) of the failed connection ($w_i^{G_1}$) then the recovery of $w_i^{G_1}$ proceeds on $b_1^{i,1}$ (regardless of the order of arrival of $FBM(f_2)$ to the source node of $w_i^{G_1}$).

B Otherwise (i.e., f_2 affects $b_1^{i,1}$), three different cases should be considered according to the arrival time of $FBM(f_2)$ at the source node of $w_i^{G_1}$ (we assume $w_i^{G_1}$, $b_1^{i,1}$ and $b_1^{i,2}$ routed through [A-B-C-J], [A-D-E-J] and [A-F-G-H-I-J] accordingly, Figure 3.3):

i Node A receives $FNM(f_1)$ after receiving $FBM(f_2)$: the source node restores the connection by sending a failure recovery message (FRM_1) along $b_1^{i,2}$ (see Figure 3.3(a)).

ii The second failure (f_2) occurs while attempting a recovery along $b_1^{i,1}$: the source node will receive $FBM(f_2)$ shortly after sending FRM1 along $b_1^{i,1}$. In this case, the source upon receiving $FBM(f_2)$ attempts a new recovery along $b_1^{i,2}$ and any reserved resources along $b_1^{i,1}$ will be released (Figure 3.3(b)).

iii The second failure occurs shortly after setting up $b_1^{i,1}$, i.e., $ACK(f_1)$ arrives earlier than $FBM(f_2)$: $b_1^{i,1}$ will be considered as a new connection and its source node is notified and restoration takes place along $b_1^{i,2}$ (Figure 3.3(c)).

3.4.2 Scheduled Recovery Procedure, Case II

As stated previously in section 3.3, the scheduling-based restoration procedure achieves successful recovery by accurately estimating the size of the surge of restoration messages arriving simultaneously at a *conflict* node(s) along the backup paths and computing the offset times accordingly (Algorithm 1). Now if a second failure occurs, recovery of one group of failed connections will interfere with the recovery of the second group, potentially creating new conflict nodes and/or increasing (changing) the size of the surge of restoration messages arriving at conflict nodes. As a result, the offset times computed for connections in both groups will yield erroneous recovery. One simple solution is to allow only one group (G_1) at a time to recover while the second group (G_2) will have the restoration of its connections shifted until connections in G_1 entirely complete their recovery.

However, some connections in G_2 may have started their recovery[1] as soon as the source nodes receive failure notification $(FNM(f_2))$ given that $FBM(f_1)$ from G_1 has not yet been received. In this case, a node along a protection route may receive recovery messages from connections in both groups and accordingly these messages must be segregated by that node (based on the detection time of the corresponding failure, which is transmitted along with the message) so that recovery messages from G_2 are processed upon the completion of all recovery messages for connections in G_1 at that node.

Therefore, to ensure a proper contention-free restoration of connections $w_i^{G_2}$, we need to determine two values: (1) the time at which all connections in failed group $(G_k, k = 1, 2)$ are expected to complete recovery (that is the maximal restoration time, $RT_{MAX}^{G_k}$, of the group); (2) the *shifted* time of G_2. This shift is intended only for the scheduling of transmission of backup data; i.e., while recovery is triggered immediately by the source node of a failed connection in G_2, the backup traffic is transmitted only when resources are successfully configured.

Derivation of $RT_{MAX}^{G_k}$

Let ψ be the number of ordered recovery messages arriving at node j for connections $w_i^{G_k}, w_{i+1}^{G_k}, \ldots, w_{i+\psi}^{G_k}$ and let $t_{i+\psi}^j$ be the arrival time of the recovery message of the last connection, that is $w_{i+\psi}^{G_k}$. The time at which the switch for this last connection is configured is t_{finish}^j:

$$t_{finish}^j = t_{i+\psi}^j + L^j_{w_{i+\psi}^{G_k}} * (T_P + T_{SC}); \qquad (5)$$

[1] By recovery here we mean the configuration of restoration resources, not the actual transmission of backup data.

Where $L^{j}_{w^{G_k}_{i+\psi}}$ is the offset time parameter of connection $w^{G_k}_{i+\psi}$ at node j, and it is derived in section 3.3. The time at which group G_k completes its recovery can then be computed as follows:

$$RT^{G_k}_{MAX} = \max_{x\in\Delta}(t^{j}_{finish}); \tag{6}$$

where Δ is the set of all nodes involved in the recovery (see Algorithm 1)

A notification message $(FNM(T^{2}_{OT,w^{G_k}_{i}}, RT^{G_k}_{MAX}, N_{G_k}, T^{G_k}_{D}, f_k))$ will be then sent to the source node of a failed connection where N_{G_k} is the total number of failed connections in group $G_k, k = 1,2$, $T^{G_k}_{D}$ is the failure detection time of that same group and f_k is the identity of the failed link. Also, the upstream node of the failed link will broadcast a failure message $FBM(RT^{G_k}_{MAX}, N_{G_k}, T^{G_k}_{D}, f_k), k = 1,2$ in the network to all nodes.

Derivation of $t^{G_2}_{shift,w^{G_2}_{i}}$.

There are two possible different scenarios depending on which message (*FBM* or *FNM*) is received first at the source node of $w^{G_2}_{i}$:

A If $FBM(f_1)$ is received before $FNM(f_2)$ and if $t^{w^{G_2}_{i}}_{f_2}$ is the arrival time of $FNM(f_2)$ at the source node of $w^{G_2}_{i}$:

$$t^{G_2}_{shift,w^{G_2}_{i}} = \begin{cases} RT^{G_1}_{MAX} - t^{w^{G_2}_{i}}_{f_2} & \text{if } RT^{G_1}_{MAX} > t^{w^{G_2}_{i}}_{f_2}; \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

B If $FBM(f_1)$ is received after $FNM(f_2)$ and $t^{w^{G_1}_{i}}_{f_1}$ is the arrival time of $FBM(f_1)$ at the source node of $w_i^{G_2}$:

$$t^{G_2}_{shift,w_i^{G_2}} = \begin{cases} RT^{G_1}_{MAX} - t_{f_1}^{w_i^{G_1}} & \text{if } RT^{G_1}_{MAX} > t_{f_1}^{w_i^{G_1}}; \\ 0 \quad , & \text{if } RT^{G_1}_{MAX} < t_{f_1}^{w_i^{G_1}} \text{ or } RT^{G_1}_{MAX} < t_{f_2}^{w_i^{G_2}} . \end{cases} \tag{8}$$

Now, because of the random nature of the failures and their occurrence in the network, notifications of the first failure may arrive *after* the broadcast failure messages of the second; moreover, some connections in G_1 may have their primary backup routes affected by the second failure. In this case, those connections will be restored on their second backup routes. However, the source nodes of these failed connections (connections in G_1 whose backup paths are affected by the second failure) have no further information to appropriately schedule their restoration. Moreover, the restoration of these connections should not affect or cause any perturbation to the already scheduled connections by creating new conflict nodes or adding new messages (altering the size of the surge of messages previously estimated) to existing conflict nodes. For this reason, we propose that connections unaffected by the second failure to be restored using the computed offset times (i.e., using the method of section 3.3). However, failed connections in G_1 whose primary backup routes are affected by the second failure will be restored only upon complete recovery of all other connections in G_1 and G_2. The new offset times for these connections are adjusted locally at their source nodes to $T^1_{OT}(N_{G_1}) + T^1_{OT}(N_{G_2})$[2] (after a source node had received both $FNM(f_1)$ and $FBM(f_2)$. This upper bound offset time will guarantee a contention free and a successful recovery for these connections.

[2] $T^1_{OT}(N_{G_k}) = \sum_{i=1}^{n-1}(N_{G_k} \times T_p^i) + T_p + N_{G_k} \times T_{SC}$

Below we describe the complete simple steps used for double failure recovery:

1. For $w_i^{G_1}$:

A If the source node receives both messages, but $FBM(f_2)$ prior to $FNM(f_1)$:

i If $b_1^{i,1}$ is affected by f_2, then the source node cannot use $T^2_{OT,w_i^{G_1}}$ as offset time. Rather, the backup traffic is restored on $b_1^{i,1}$ and the offset time to be used is $T^1_{OT}(N_{G_1}) + T^1_{OT}(N_{G_2})$.

ii Otherwise, $T^2_{OT,w_i^{G_1}}$ is used to offset the retransmission of backup data along $b_1^{i,1}$.

B Otherwise, (i.e., the source node of $w_i^{G_1}$ has received $FNM(f_1)$ but not $FBM(f_2)$), therefore it triggers the recovery procedure of its failed connection(s) using the computed offset time(s), irrespective of whether another failure had occurred or not (since no FBM message had been received). Later, when $FBM(f_2)$ is received:

i If $b_1^{i,1}$ is affected by f_2, then the source node initiates a new recovery along $b_1^{i,2}$ and it uses a new offset time $T^1_{OT}(N_{G_1}) + T^1_{OT}(N_{G_2})$ to restore its traffic. Any reserved resources along $b_1^{i,1}$ will be released.

ii Otherwise, ignore the received message.

2. For $w_i^{G_2}$:

A If the source node receives both messages, but $FBM(f_1)$ prior to $FNM(f_2)$:

i If $b_2^{i,1}$ is affected by f_1 , the source node commences recovery on $b_2^{i,2}$ and schedules its backup retransmission using the upper bound offset time, T^1_{OT} (NOTE:

here, the computed offset time ($T^2_{OT,w_i^{G_2}}$) cannot be used along $b_2^{i,2}$ since it was computed for restoration along $b_2^{i,1}$, however, the upper bound will guarantee successful recovery as it assumes worst case delays among connections in G_2) after shifting by $t^{G_2}_{shift,w_i^{G_2}}$(to guarantee that G_1 has completed its recovery). Therefore, the new offset time for $w_i^{G_2}$ is $t^{G_2}_{shift,w_i^{G_2}} + T^1_{OT}(N_{G_2})$.

ii otherwise, $w_i^{G_2}$ is restored on $b_2^{i,1}$ and the new offset time is: $t^{G_2}_{shift,w_i^{G_2}} + T^2_{OT,w_i^{G_2}}$.

B Otherwise, (i.e., the source node of $w_i^{G_2}$ receives $FNM(f_2)$ first), it immediately starts its recovery (as no information is available about the first failure) along its primary backup path $b_2^{i,1}$. Upon receiving $FBM(f_1)$:

i If $b_2^{i,1}$ is not affected by f_1, it schedules its backup data retransmission using a offset time: $t^{G_2}_{shift,w_i^{G_2}} + T^2_{OT,w_i^{G_2}}$.

ii Otherwise, it will release any reserved resources along $b_2^{i,1}$ and restarts its recovery along $b_2^{i,2}$ and reschedules its backup data retransmission using a new offset time: $t^{G_2}_{shift,w_i^{G_2}} + T^1_{OT}(N_{G_2})$

3.5 Performance Evaluation

We evaluate the performance of the proposed recovery scheme and we compare it with the conventional two-way messaging restoration procedure. The 16-node NSF network [1] is used in our simulation study. The metric for the performance evaluation is the network restoration time. An event-driven simulation tool is developed to model the distributed

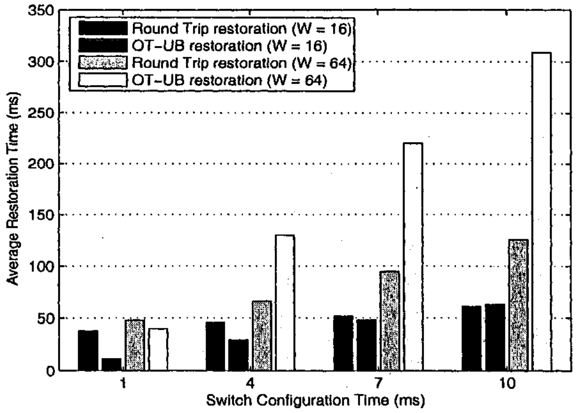
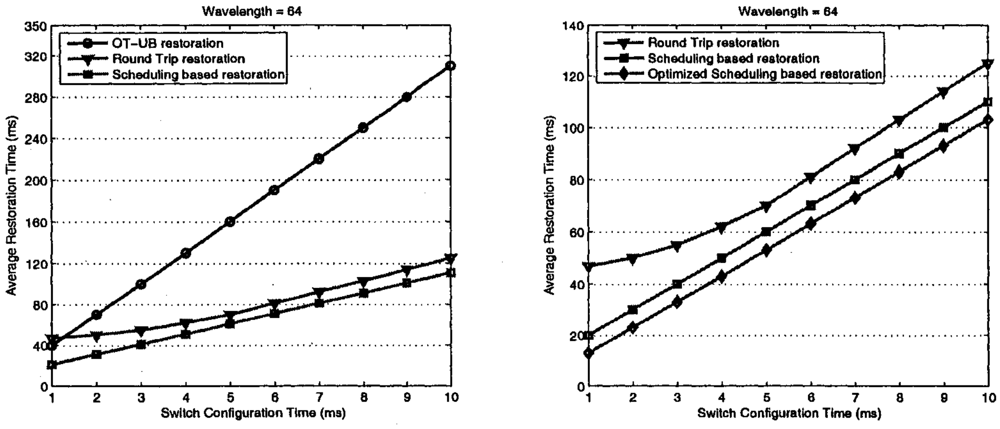


Figure 3.5: Network RT vs. T_{SC}

provisioning and recovery protocol. We assume that each link consists of two unidirectional fibers. Connection requests arrive as a Poisson process and the connection-holding time follows a negative exponential distribution with mean $1/\mu$ = 100 ms. The Random Wavelength Selection Algorithm is used to select the candidate wavelength for setting up connections. We simulate the failure of one (or two) unidirectional link(s) after running the simulation for some period and satisfying some traffic demand.

3.5.1 Performance under Single Link Failure

We start by comparing the two-phase conventional restoration and the OT-UB restoration schemes. Figure 3.4 shows the performance of the two restoration protocols under different setting parameters. Clearly, when the number of wavelength is smaller and/or the switch configuration time is shorter, the OT-UB scheme outperforms the two-phase messaging approach (as the impact of N, number of failed connections, is minimal). But as the switch configuration time or the number of wavelength per link increases, the performance of the OT-UB degrades. The reason that the OT-UB degrades faster (i.e., larger slope) is due to



(a) (b)

Figure 3.6: Network RT vs. T_{SC}, $W = 64$

the fact that the scheme considers all failed connections (which may not necessarily be in contention state) and more switching delays start to have greater impact on the restoration times. This obviously results in an overestimation of the offset times, which further diminishes the performance of OT-UB rendering the argument of avoiding the conventional two-phase signaling inappropriate. On the other hand, the conventional two-way messaging is slightly affected since the round trip propagation delays are the major factor affecting its performance.

Clearly, overestimating the number of restoration messages arriving at conflict nodes results in performance degradation. Therefore, the proposed scheduling scheme provides a closer approximation of the size of the contending messages and models the sequence of their arrival at intermediate nodes along the restoration routes. Figure 3.5(a) shows a comparison among the conventional two-phase restoration, OT-UB restoration, and the

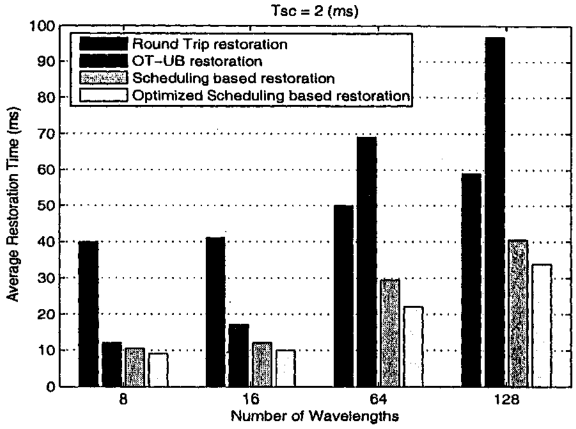


Figure 3.7: RT vs. number of wavelengths, $T_{SC} = 2ms$

scheduling-based restoration for 64 wavelengths. As the figure shows, substantial improvement can be achieved over the OT-UB by avoiding the use of upper bounds for the OT, whereas a moderate improvement (30ms - 15ms) can be achieved over the conventional two-way messaging approach. The figure also shows that as T_{SC} increases and/or the number of wavelengths increases, the new scheduling recovery framework is not affected and its slope is consistent with that of the two-way messaging. This suggests that we modeled quite accurately the size of the surge of messages contending at conflict nodes while also eliminating the round trip delays.

As we mentioned earlier, some messages may arrive consecutively at a conflicting node and depending on their interarrival time, they may or may not be in conflict. We proposed a more optimal scheduling scheme to account for these cases, and here we show its performance in Figure 3.5(b). Clearly, by tracing the interarrival time of restoration messages we are able to better and more accurately estimate the number of conflicting messages and therefore determine exactly the OT a connection needs to wait before it restores. Here, an improvement of 35ms-20ms is achieved over the two-phase signaling.

We can also study the impact of the number of wavelengths on network recovery times. Figure 3.6 shows the network restoration times for different wavelengths when $T_{SC} = 2ms$. It is interesting to see that the multiplier of T_{SC} in the expression of the OT for the OT-UB has larger impact on the network restoration time especially as W increases. This reveals again the negative impact of the overestimation that OT-UB yields over the other schemes. Whereas the proposed scheduling scheme shows slow increase that is consistent with the conventional messaging protocol, which backs our argument in saying that the proposed restoration parameter accurately models the surge of restoration messages flow. The figure shows that when W is small, even the OT-UB scheme performs better than the conventional two-way messaging; however, its performance (i.e., the OT-UB) greatly depends on the network parameters. Note that the new proposed approach will exhibit good perform under varying network conditions.

Moreover, we study the impact of the switch architecture on the network recovery times. Namely, we compare the two-way messaging procedure and the scheduling-based restoration for both the consecutive and the parallel switching fabric [49]. Figure 3.7 shows the simulation comparison. Clearly, parallel switch architecture eliminates the switching delays and therefore results in shorter recovery times. The only delays this architecture incurs are the fault notification delays and the round-trip delays of the two-way messaging recovery. The latter delay is clearly eliminated when scheduling based restoration is deployed and in comparison with the two-way recovery scheme, restoration times of 10-20 ms can be achieved.

Finally, we study the impact of the propagation delays on the restoration procedure. As stated before, OT based restorations eliminate completely the impact of these delays.

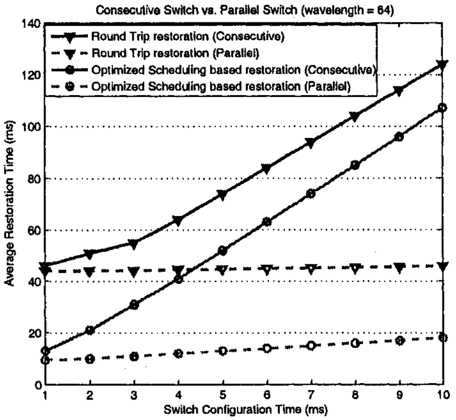


Figure 3.8: Consecutive vs. Parallel comparison

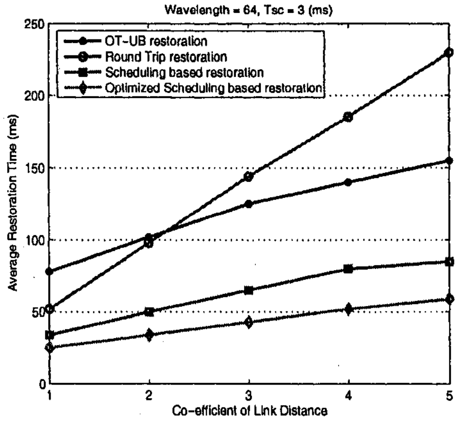


Figure 3.9: *RT* vs. link distance coefficient (consecutive)

Now to study the impact of the propagation delays we use the NSF network with virtual link distances. A new multiplier coefficient is introduced to linearly increase the distances between two adjacent nodes; e.g., a coefficient of 2 will double the links distances. Figure 3.8 shows the simulation results when the number of wavelengths is 64 and T_{SC} = 3ms. As the distances increase, the propagation delays increase and the restoration time of the conventional two-way signaling substantially increases whereas this effect on the scheduling restoration scheme is negligibly small.

3.5.2 Performance under Double Link Failures

In this section we evaluate and compare the effectiveness of the Round-Trip restoration and the optimized scheduling based restoration under the double-link failures assumptions. We let β be the inter-failure time of the two consecutive failures. There is an interesting behavior for the Round-Trip scheme (Figure 3.9). When β is small, the *RT* increases; this is justified by the fact that while connections in G_1 are being restored, the second failure occurs and affects some of those connections before completing their recovery. Therefore, they stop their recovery on their first backup routes and schedule a new recovery on their second backup routes. Now as β slowly increases, some of the connections in G_1 will be given more time to do their recovery but not enough to complete it; when the second failure occurs, these connections will be restored along their second backup paths and hence there is an increase in the RT. When β is large, some of the connections from G_1 affected by the second failure will have enough time to completely finish the recovery and therefore, when the second failure occurs, they will be recovered as new connections in G_2. Hence, we notice a decay in the restoration time as β increases (note that the RT now does not account for the attempt being made along the first primary backup).

Alternatively, scheduling recovery exhibits slightly different behavior. First, note that some connections in G_1 are affected by f_2 and they are recovered upon the completion of all connections in G_1 and G_2. Second, when β is small, $RT^{G_1}_{MAX} - t^{w_i^{G_2}}_{f_2}$ is large and therefore, the shift time for $w_i^{G_2}$ is large and the RT is large. As β increases, subsequently, connections in G_2 will be shifted only for a small period. Therefore, the restoration time decreases as the inter-failure arrival time increases. Note that the Round-Trip scheme exhibits better performance than the scheduling scheme when β is small; that is due to the fact that when both failures occur near-simultaneously, the scheduling scheme (1) recovers using OT-UB and moreover, (2) the shift period for connections in G_2 is bigger.

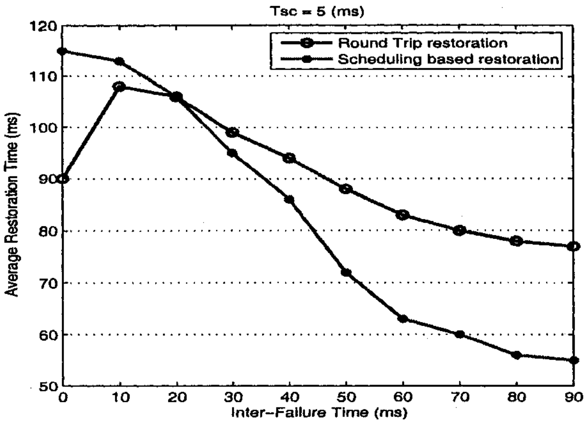


Figure 3.10: *RT* vs. Inter-failure time (β)

3.6 Conclusion

In this chapter, we addressed a fundamental problem for designing a survivable optical transport networks; that is the capability to quickly recover from element failures. By using the framework of Offset-Time and a time-driven scheduling procedure, we proposed an accurate model to estimate the offset time of each failed connection. Comparing with the Round-Trip restoration and OT-UB restoration, we showed that our proposal eliminates the propagation delays and the accumulation of switching delays, and it achieves the best recovery performance upon the single link failure. We also studied the applicability of the proposed recovery protocol under double link failures assumption. Through detailed simulation experiments, we showed that by deploying this scheduling scheme, substantial restoration gain can be achieved under varying network conditions.

Chapter 4

Improving the Robustness of Shared Network against Multiple Failures through Capacity Reconfiguration

4.1 Introduction

Significant progress has been made towards making optical networks resilient in the event of single link failures by various protection and restoration schemes. As the size and the complexity of optical mesh networks continue to grow, dual failures become increasingly probable. Double link failures can dramatically disrupt the services offered by the network if appropriate precautions are not implemented. Hence designing recovery algorithms to protect against such failure events and to ensure service continuity is a paramount concern. To date, various research efforts [54, 55, 52, 53] have already addressed the problem of routing connections under dual failure assumptions, and findings show that designs offering complete dual-failure restorability require more than double the amount of spare capacity [54, 55].

In order to avoid this excessive deployment of extra spare capacity in the network to achieve higher dual failure restorability, capacity reprovisioning/reconfiguration[1] after the occurrence of and recovery from the first failure has been proposed [56, 57, 58, 59, 60]. After the occurrence of the first failure, the failed connections are restored from their working paths into their protection paths. Therefore, upon a complete recovery, shared protection capacity along now active protection routes can no longer be shared. As a result, some of the connections in the network that are not directly affected by the failure will become unprotected (or exposed) and will dramatically increase the network vulnerability to a subsequent failure. Spare capacity reconfiguration provides a mechanism by which one can find and re-allocate new protection capacities for these newly unprotected connections without a priori knowledge of the location of the second failure.

In this chapter we study the benefits of capacity reprovisioning after a failure, particularly on improving the robustness of connections in shared optical mesh networks. We assume two independent link failures, where the second failure occurs after the first failure is recovered from, but before it is physically repaired. A critical objective for reprovisioning is to reduce the total number of connections that have to be re-provisioned. Here the motivations are twofold: (1) to reduce management overheads in simultaneously provisioning a large number of connections, and (2) to lower reservation contention between

[1] In this thesis, the terms "reprovisioning" and "reconfiguration" are used interchangeably. Reprovisioning does not mean a new capacity is placed into the network.

multiple unprotected connections trying to establish backup capacity. The latter may result in increased blocking rates during reconfiguration, which in turn will increase the number of exposed connections and hence the vulnerability to subsequent failure(s).

We present and compare the performance of two different reprovisioning schemes under both centralized and distributed implementations. We reconfirm that reprovisioning mitigates the impact of double link failures and dramatically improves the network robustness in a network that is only designed to achieve 100% restorability under single link failures. Moreover, we show that the new scheme proposed here outperforms a conventional scheme [60]; that is under the same failure circumstances, our scheme re-provisions fewer connections than the conventional approach (i.e., reduced overhead and contentions) and protects more (i.e., better robustness). We also show that under distributed implementation the performance of both reprovisioning schemes degrades. Subsequently, we present a simple technique to cope with the adverse effects of contentions by allowing unsuccessful connections to reattempt reprovisioning [71]. We show that reattempting substantially improves the network performance when distributed reconfiguration is implemented.

The rest of Chapter 4 is organized as follows. In section 4.2 we discuss the prior work in the literature for handling multiple link failures. We introduce the lightpath reprovisioning problem and we present some analysis in section 4.3; we propose a new algorithm for capacity reconfiguration in section 4.4 and present its analysis. Section 4.5 presents the centralized and distributed implementation of the reconfiguration algorithm. Section 4.6 presents performance evaluation and finally we conclude this chapter in section 4.7.

4.2 Related Work

Various research efforts[52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66] have addressed the problem of routing connections under dual failure assumptions.

The authors of [52] analyzed the spare capacity requirements for a network employing link protection to provide complete double-link failure recovery. Two edge-disjoint backup paths were computed for each link for rerouting traffic when a pair of links fails. Numerical results suggested that it is possible to achieve almost 100% recovery from double-link failures with a moderate increase in backup capacity. The authors also showed that it is possible to trade off capacity for restorability by choosing a subset of double-link failures and designing backup paths for only those failure scenarios. The work presented in [53] extended the study in [52] to include backup capacity sharing that improves resource utilization and also formulated an ILP towards the same. The authors of [55] considered span restorable mesh networks and evaluate the restorability when double link failure occurs. They showed that networks originally designed to withstand single span failures can inherently restore about 70% of failed capacities in double failure scenario. Furthermore the authors found that in a modular capacity environment adaptive span restoration process can restore as much as 95% of failed capacity on average over all dual failure scenarios [61]. In [54], the authors presented different design strategies to achieve high dual failure restorability. They noted that achieving complete dual failure restorability yields extremely high capacity penalty. A design model that allows a user to specify a capacity budget and hence obtain the highest average dual failure restorability was presented. Additionally, the authors proposed another design model that supported multiple restorability service classes at remarkably low cost within span restorable mesh networks that were already efficiently designed to only support single failure restorability. The authors of [65] studied the effect of span maintenance actions on network restorability. An optimization method was introduced to allow for a controlled tradeoff between the cost of spare capacity and the reduction of restorability risk. In [62] the authors provided methods for the analysis of dual failure restorability and related availability considerations is shared backup path protection (SBPP). An integrated strategy was discussed: first failure protection, second failure restoration. They showed that in a network nominally designed to withstand single failures, dual failure restorability could be enhanced through spare capacity sharing limits for SBPP. The authors of [64] addressed the problem of multiple failures where a combination of path level protection and link layer restoration were proposed to deal with the first and the second failure respectively.

On the other hand, dynamic protection reconfiguration was discussed in [57] where spare capacity was reconfigured after the first failure. Results showed that reconfiguration can be implemented with as little as 4% additional capacity. The authors of [56] studied the restorability performance and capacity consumption of path protection and rerouting mechanisms in networks designed for single and double link failures. Recently, a backup reprovisioning algorithm for handling multiple failures was presented in [60] and comprehensive studies indicated that spare capacity reconfiguration can dramatically lower network vulnerability. Similarly, others have considered pre-emptive network reprovisioning schemes [58] whenever the second failure is assumed to occur after recovery actions are taken for the first failure but before the actual failed link itself is restored; overall findings showed a notable improvement in the level of network vulnerability as well as recovery ratios. The work of [63] proposed methods for achieving high dual failure restorability in p-cycle networks that are optimally designed only to withstand all single failures or have minimized amounts of additional capacity for dual failure considerations. The authors considered the cases of both static p-cycles and reconfigurable p-cycles.

In this work we assume shared mesh networks that are designed to protect only against link single failures. We propose to enhance the dual failure restorability of these networks through spare capacity reconfiguration upon the occurrence of a failure; capacity reconfiguration identifies exposed connections after a failure and attempts at provisioning new spare capacity to protect them. We present a new reconfiguration method and compare its performance to previous ones. To evaluate these methods, we assume in our study two independent near simultaneous failure where the second failure occurs right after recovery from the first one.

4.3 Lightpath Reprovisioning

4.3.1 Background

In protected optical mesh networks, lightpath provisioning requires the computation and establishment of end-to-end working and link-disjoint protection paths. Our study herein strictly focuses on shared protection since one of the key benefits of shared protection in mesh networks is the elimination of capacity overbuild, generally associated with dedicated protection.

Now under normal conditions, the network is usually protected against all single failures. Namely, when a failure occurs, all connections whose working paths are affected by that failure are re-routed on their corresponding protection paths [30, 45, 47]. However, since these protection resources may also be shared with other unaffected connections, these may become unprotected and vulnerable to the next failure [60]. Figure 4.1 shows an illustrative example with three connections (A-H, C-G, and D-F). A working lightpath (w_i) and a physically disjoint backup lightpath (b_i) for each connection are initially provisioned. The protection resource on link (D-E) is shared between b_1 and b_2 since their corresponding working paths (w_1 and w_2) are link-disjoint. Upon the failure of link (B-F), all working connections that are routed through that link are re-routed onto their corresponding protection paths. This in turn yields a set of unprotected connections, which increases the vulnerability to a second failure. Overall, to summarize these unprotected connections, one can classify them into three categories:

1) *Directly Affected Working lightpaths*: A failed demand that is re-routed to its backup path is still vulnerable to a second failure that may affect its active protection path (e.g., b_1).

2) *Directly Affected Backup lightpaths*: Demands whose protection connections have failed due to the first failure (e.g., w_3).

3) *Indirectly Affected lightpaths*: Upon failure, shared protection resources are activated by the failed connections which may cause some connections (whose backup lightpaths share these protection resources) to become unprotected (e.g., w_2). It is also called *Vulnerable Connections*.

Clearly, a higher number of unprotected connections after the recovery from a failure

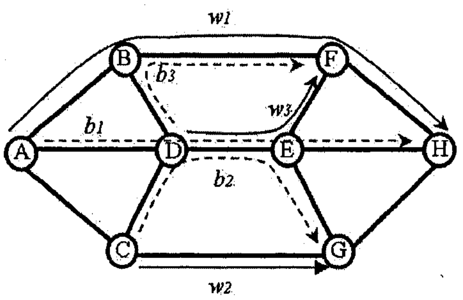


Figure 4.1: Sample Network and Connections

can increase the vulnerability of the network to subsequent failures and therefore lower its overall restorability. To improve the service availability, reprovisioning or capacity reconfiguration exploits the available capacity in the network to re-establish new backup paths for unprotected connections in advance of a failure (and right after the recovery from the first fault). Unlike dynamic restoration, reprovisioning can improve network availability by markedly reducing service unavailability times, e.g., from hours to tens of seconds [58, 59].

4.3.2 Re-provisioning Analysis

A reprovisioning algorithm typically takes several inputs: 1) the network topology along with state information (i.e., resource availability/sharability, existing working and protection routes, protection statuses, etc), and 2) a list U of unprotected demands or demands that require reprovisioning. The algorithm then tries to establish backup lightpaths for unprotected connections using available capacity in the network. One such algorithm has been proposed in [60] and its performance is evaluated here as well, termed thereafter as Scheme I. The details of the algorithm are now discussed after the introduction of necessary notations:

W : The total number of wavelengths per link;

l : The failed link;

$\{w_1^l, w_2^l, \ldots, w_x^l\}$: The set of working lightpaths going through link l, with corresponding backup lightpaths $b_1, b_2, \ldots, b_x$, respectively;

$\{b_1^l, b_2^l, \ldots, b_y^l\}$: The set of backup lightpaths using protection wavelengths along link l;

$\{l_1^{b_i}, l_2^{b_i}, \ldots, l_m^{b_i}\}$: The set of physical links along the route of a backup lightpath b_i;

$B_{a,i}$=$\{b_1^{l_a^{b_i}}, b_2^{l_a^{b_i}}, \ldots, b_{k_{a,i}}^{l_a^{b_i}}\}$: The set of backup lightpaths sharing the same wavelength λ with backup lightpath b_i on link $l_a^{b_i}$, $a = 1, \ldots, m$, whose corresponding working lightpaths are $U_{a,i}$=$\{w_1^{l_a^{b_i}}, w_2^{l_a^{b_i}}, \ldots, w_{k_{a,i}}^{l_a^{b_i}}\}$;

Note that when link l fails, connection w_i^l is restored to its backup path b_i and as a result protection capacities of $\{b_1^{l_a^{b_i}}, b_2^{l_a^{b_i}}, \ldots, b_{k_{a,i}}^{l_a^{b_i}}\}$ on link $l_a^{b_i}$ become unavailable; thereby leaving demands $\{w_1^{l_a^{b_i}}, w_2^{l_a^{b_i}}, \ldots, w_{k_{a,i}}^{l_a^{b_i}}\}$ unprotected. Similarly, each link $l_a^{b_i}$, $a = 1, \ldots, m$, along the route of backup lightpath b_i will yield a set of unprotected demands. Let S_i be the set of all unprotected demands resulting from the recovery of w_i^l, whose size is δ_i:

$$\delta_i = |S_i| = k_{1,i} + (k_{2,i} - \psi_{2,i}) + (k_{3,i} - \psi_{3,i}) + \cdots + (k_{m,i} - \psi_{m,i}) \tag{9}$$

where, $k_{a,i}$ is the total number of unprotected connections resulting from the activation of the shared capacity (wavelength) on link $l_a^{b_i}$ (by active backup lightpath b_i), $k_{a,i}$=$|U_{a,i}|$. $\psi_{a,i}$ is the number of unprotected connections resulting from the activation of the shared capacity on link $l_a^{b_i}$ by active backup lightpath b_i *and* already became unprotected on any of the upstream link(s) of b_i, namely $\{l_1^{b_i}, l_2^{b_i}, \ldots, l_{a-1}^{b_i}\}$. The following equations can be derived:

$$\psi_{1,i} = 0; \tag{10}$$

$$0 \leq \psi_{a,i} \leq k_{a,i}; \tag{11}$$

$$\psi_{a,i} = |B_{a,i} \cap B_{a-1,i}|, a = 2, \ldots, m; \tag{12}$$

Therefore, the number of unprotected connections resulting from the recovery of w_i^l (i.e., *indirectly affect connections*):

$$\delta_i = k_{1,i} + \sum_{a=2}^{m} (k_{a,i} - \psi_{a,i}) \leq \sum_{a=1}^{m} (k_{a,i}) \tag{13}$$

Hence, the total number of unprotected connections, Δ, that results from the recovery of all failed working connections (w_i^l,i=1,...,x) into their respective backups:

$$\Delta = \delta_1 + (\delta_2 - \pi_2) + (\delta_3 - \pi_3) + \cdots + (\delta_x - \pi_x) \tag{14}$$

where $\pi_i (0 \leq \pi_i \leq \delta_i, i = 2, \ldots, x)$ represents the number of resulting unprotected connections in the set S_i that were also unprotected in other sets, i.e., $S_1, S_2, \ldots, S_{i-1}$. For example, π_2 is the number of connections in the network that become unprotected after the recovery of w_2^l and are also unprotected upon recovery of w_1^l (i.e., $\pi_2 = |S_1 \cap S_2|$). Hence, let X_2 represent the set of unprotected connections that result from the recovery of w_2^l and are not counted in S_1; i.e., X_2=$\{x \in S_2 \mid x \notin S_1\}$ and $\delta_2 - \pi_2$ is its size. Similarly, X_3=$\{x \in S_3 \mid x \notin S_2 \wedge x \notin S_1\}$ and X_i=$\{x \in S_i \mid x \notin S_{i-1} \wedge \cdots x \notin S_2 \wedge x \notin S_1\}$, and therefore one can simply write:

$$\delta_2 - \pi_2 = |X_2| = |S_2| - |S_2 \cap S_1| = |S_2 \cap \bar{S}_1| = |S_2 \cup S_1| - |S_1| \tag{15}$$

$$\delta_3 - \pi_3 = |X_3| = |S_3 \cap \bar{S}_2 \cap \bar{S}_1| = |S_3 \cup S_2 \cup S_1| - |S_2 \cup S_1| \tag{16}$$

$$\delta_i - \pi_i = |X_i| = |S_i \cap \bar{S}_{i-1} \cap \cdots \cap \bar{S}_1| = |S_i \cup S_{i-1} \cup \cdots \cup S_1| - |S_{i-1} \cup \cdots \cup S_1| \tag{17}$$

where $\bar{S}_i$ is the complement set of S_i. Finally, the total number of unprotected connections resulting from the failure of link l is:

$$\Delta_l = \Delta + x + y - Y = \sum_{i=1}^{x} (\delta_i - \pi_i) + x + y - Y, \pi_1 = 0 \tag{18}$$

where x is the number of failed working connections and y is the number of failed backup connections (in category 2, see section 4.3.1), Y is the number of failed backup connections that have already been counted in Δ $(0 \leq Y \leq y)$. In other words, a connection whose failed backup lightpath shares protection capacity with the backup lightpath (of a failed connection) needs to be excluded when counting the unprotected connections resulting from the failure of since it has been considered and counted by.

Once the set of all unprotected connections is identified, a reprovisioning algorithm is usually triggered to provision backup capacity for unprotected demands. The steps of Scheme I approach are detailed in Algorithm 2. We note here that the performance of the reprovisioning algorithm is order dependent. The authors of [60] presented different schemes for the order of connection reprovisioning; namely, the random, the longest backup and the most violations policies. In this work, the random policy is used for its simplicity.

Algorithm 2 Pseudo code of Scheme I in Reprovisioning

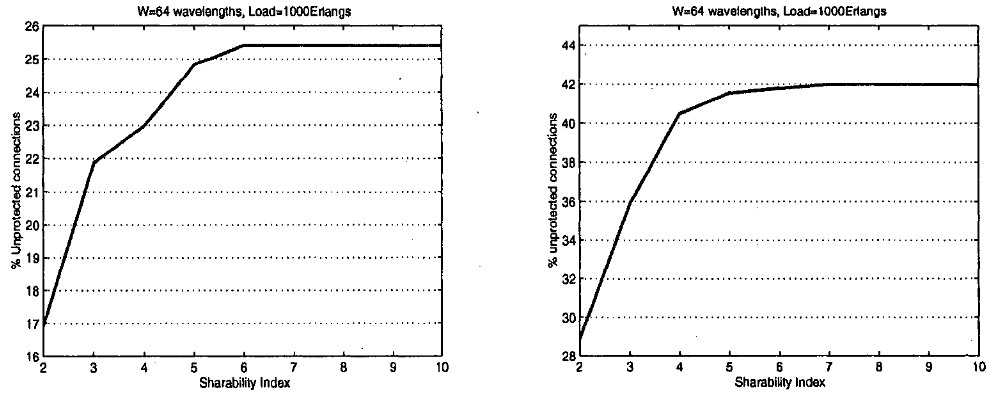
```
1: Re-route each demand whose working path is affected by the first failure onto its backup route, release resources along the failed working path.
2: Identify the list of unprotected connections (based upon the above discussion):
3: For each unprotected demand, release the protection capacity that has already been reserved but can no longer be used. Repeat until all demands are processed.
4: Compute a link-disjoint route for the working path of each unprotected demand and allocate protection capacity if available.
5: Reserve capacity and go back to step 2.b. Repeat until all unprotected connections are processed.
```

4.3.3 Impact of Resource Sharability

The level of sharability of network protection resources will have a strong impact on the network vulnerability and the performance of reprovisioning algorithms. A higher sharability implies that more demands are admitted to the network since more backup connections are packed together over the same protection capacity; however, as a failure occurs and failed connections are restored into their protection paths, a larger number of demands become unprotected (the impact of failed connections in category 2 and 3) and therefore vulnerable to subsequent failures. Figure 4.2 shows a study on a nation wide network (Figure 4.4) where the percentage of unprotected connections in measured after the recovery from the first failure by varying the sharability levell (or sharability index) of protection resources. Clearly, the figure shows a noticeable increase in the number of unprotected connections (e.g., 9% increase in wavelength continuous network and 13% in a wavelength convertible network) in the network after the first failure if unlimited resource sharability is allowed. Therefore, intuitively this finding suggests that limiting the level of sharability will result in a lower number of unprotected demands after recovery and therefore in a less number of connections to be reprovisioned [62].

However, on the other hand, lower level of sharability does not suggest that the percentage (or the number) of unprotected connections in the network after reconfiguration is reduced. The reason for that is that limited resource sharability will effectively limit the flexibility of the reprovisioning algorithm in finding protection resources for unprotected demands. Another side effect is that limited sharability will limit the overall performance of the network since fewer connections can be accommodated by the network. Alternatively, unlimited resource sharability may provide the necessary flexibility in allocating protection capacity to unprotected demands after the first failure but as the figure shows a larger group of demands become unprotected and require reprovisioning. Moreover, if distributed reprovisioning is implemented, larger number of connections attempting to reserve protection capacity will amplify the contentions over resources yielding higher blocking and leaving more connections unprotected even when resources are available.

Therefore, it is clear that there are two conflicting design constraints: on one hand limited level of sharability may reduce the number of unprotected connections but at the expense of limited network performance and less flexibility in allocating protection capacity for unprotected connections; on the other hand, higher level of sharability may result in larger number of unprotected connections after the first failure and magnifies the effect of contentions under distributed control but yields higher degree of flexibility in provisioning protection capacity.



(a) No wavelength conversion

(b) Full wavelength conversion

Figure 4.2: Percentage of Unprotected Connections before Re-provisioning vs. SI

4.4 A New Reprovisioning Approach

In the previous section, we presented a scheme for resolving the list of unprotected connections that result from link failure recovery. We showed that upon the recovery of a failed demand (w_i^l) the shared protection capacity along its backup b_i becomes unavailable and therefore a total of connections $(\delta_i - \pi_i)$ become unprotected and require new backup reprovisioning. This number may be quite large, especially if unlimited resource sharability is allowed [67]. Therefore, the motivation to propose a new reprovisioning scheme is to reduce the total number of connections that have to be reprovisioned; thus to decrease both the management overheads in provisioning a large number of backup connections and the reservation contention (conflict) between multiple unprotected connections trying to re-establish backup capacity.

Note that when a connection (w_i^l) is restored onto its backup (b_i), shared protection capacity along b_i becomes *temporarily unavailable* for other demands whose backup routes share that capacity (i.e., $\{w_1^{l_a^{b_i}}, w_2^{l_a^{b_i}}, \dots, w_{k_{a,i}}^{l_a^{b_i}}\}$, $a = 1, \dots, m$). Instead of provisioning new backup capacity for these newly unprotected demands (whose number may be large), a new working path $w_i^{l,new}$ may be provisioned for each failed lightpath, w_i^l, that is link-disjoint with b_i. Upon successfully completing the provisioning of $w_i^{l,new}$, the traffic is simply reverted back from b_i to $w_i^{l,new}$ leaving the rest of unaffected connections intact. Here, traffic is switched back to $w_i^{l,new}$ upon successfully provisioning the required resources avoiding any disruptions. However, note that the protection capacity along b_i may not preserve its sharability status since $w_i^{l,new}$ could be non link-disjoint with (some) demands whose protection lightpaths share protection capacity with b_i (i.e., $\{w_1^{l_a^{b_i}}, w_2^{l_a^{b_i}}, \ldots, w_{k_{a,i}}^{l_a^{b_i}}\}$, $a = 1,\ldots,m$). In such a case, a new backup path (b_i^{new}) or new protection capacity along b_i needs to be provisioned to protect $w_i^{l,new}$, avoiding the disruption of $\{w_1^{l_a^{b_i}}, w_2^{l_a^{b_i}}, \ldots, w_{k_{a,i}}^{l_a^{b_i}}\}$, $a = 1,\ldots,m$. If the provisioning of $w_i^{l,new}$ (and/or b_i^{new}) fails then we compute the set of all unprotected connections resulting from the recovery of w_i^l and re-provision them accordingly (similar to scheme I). Note that when wavelength conversion is deployed, only the links along b_i where protection wavelength(s) cannot be shared are identified and new protection wavelength(s) on those links are provisioned. Finally, the same procedure is applied to all working paths $\{w_1^l, w_2^l, \ldots, w_x^l\}$ of failed demands. Upon finishing this phase (phase 1), other unprotected connections in other categories that are not considered in phase 1 are reprovisioned.

The effectiveness of this new scheme (termed thereafter as scheme II) is best shown via an illustrative example in Figure 4.3. We assume initially b_1, b_2 and b_3 are all setup using λ_1, and b_1 shares λ_1 on link (D-E) with b_2 and on link (E-H) with b_3. When link (B-F) fails, w_1 is restored to its backup b_1 and as a result, b_2 and b_3 become unavailable since they can no longer share protection capacity with b_1. Hence b_1, w_2 and w_3 become unprotected and

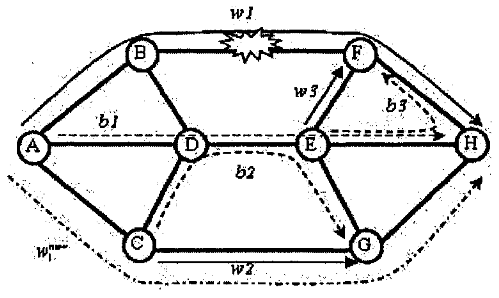


Figure 4.3: Example for Re-provisioning

three new protection paths (or capacity) need to be reprovisioned in Scheme I in order to fully protect the network against a subsequent failure. Under scheme II however, when w_1 is restored to its backup, connection b_1, w_2 and w_3 become *temporarily unprotected.* Hence, if we can find a new working path (w_1^{new}) that is link-disjoint with b_1 to carry the failed traffic, then b_2 and b_3 can become available again and their corresponding connections are fully protected. Note that w_1^{new} may not be link-disjoint with w_2 and/or w_3 (w_2 in this example). Therefore, b_1 cannot share any protection resource with b_2. In a wavelength continuous network, a new backup b_1^{new} (and protection wavelength) that is link-disjoint with w_1^{new} has to be provisioned. In a wavelength convertible network, the conflict links are identified (e.g., (D-E)) and a different wavelength is provisioned along those links (e.g., λ_2 can be assigned to b_1 on link (D-E) leaving the rest of the backup lightpath intact).

Note that this approach differs from the Scheme I. Here the total number of temporarily unprotected connections during the reprovisioning time is equal to Δ_l, and the total number of connections to be reprovisioned is computed differently. Namely, when a connection w_i^l is rerouted to its backup b_i, a total of $\delta_i - \beta_i + 1$ ($0 \leq \beta_i \leq \pi_i \leq \delta_i, i = 2, \ldots, x$; "1" means w_i^l itself) connections become temporarily unprotected. Now if the algorithm is successful in provisioning $w_i^{l,new}$ or ($w_i^{l,new}$, b_i^{new}), then $\delta_i - \beta_i$ connections do not require reprovisioning;

otherwise, $\delta_i - \beta_i + 1$ connections require reprovisioning. Here, β_i is computed in a similar way to π_i except that when a connection (e.g., w_i^l) successfully provisions $w_i^{l,new}$, then its corresponding set (S_i) becomes empty. Example, $\delta_2 - \pi_2 = |S_2| - |S_2 \cap S_1| = |S_2|$, when $w_1^{l,new}$ is successfully provisioned. The total number of connections requiring reprovisioning upon the re-routing of w_i^l can be expressed by:

$$\chi_i = \eta_i + 2(1-\eta_i)\mu_i + (1-\eta_i)(1-\mu_i)(\delta_i - \beta_i + 1) \tag{19}$$

where: $\eta_i = \{0,1\}, \mu_i = \{0,1\}, \beta_1 = 0$, and $0 \leq \beta_i \leq \pi_i \leq \delta_i, i = 2, \ldots, x$.

η_i is a parameter that is set to 1 when $w_i^{l,new}$ is successfully reprovisioned, otherwise it is set to 0. μ_i is a parameter that is set to 1 when ($w_i^{l,new}$, b_i^{new}) require reprovisioning, otherwise it is set to 0 and $\delta_i - \beta_i + 1$ connections become unprotected and require reprovisioning. Hence, the total number of connections requiring reprovisioning in this new scheme upon the first failure is:

$$\nabla_l = y - Y' + \sum_{i=1}^{x}(\chi_i) \tag{20}$$

where $Y'(0 \leq Y' \leq Y \leq y)$ is the number of unprotected connections corresponding to the failed backup connections $\{b_1^l, b_2^l, \ldots, b_y^l\}$ that have already been considered.

Now, the lower bound for ∇_l is defined when all failed connections (w_i^l) are rerouted to their backup paths (b_i) and then new working connections ($w_i^{l,new}$) are successfully provisioned. Here, $\eta_i = 1$ and $Y' = 0$, therefore:

$$\nabla_l^{lower} = x + y \tag{21}$$

On the other hand, the upper bound (i.e., when all failed connections do not succeed in provisioning $w_i^{l,new}$ (or $w_i^{l,new}$ and b_i^{new}) corresponds to $\eta_i = 0, \mu_i = 0, Y' = Y$, and $\beta_i = \pi_i$:

$$\nabla_l^{upper} = y - Y + \sum_{i=1}^{x}(\delta_i - \pi_i + 1) = x + y - Y + \sum_{i=1}^{x}(\delta_i - \pi_i) = \Delta_l \tag{22}$$

and therefore,

$$x + y \leq \nabla_l \leq \Delta_l \tag{23}$$

This clearly shows that, in Scheme II, the number of unprotected connections to be reprovisioned in order to improve the network restorability can be significantly reduced and its worst case performance converges to that of scheme I. The steps of scheme II are detailed in Algorithm 3:

As before, higher level of resource sharability leads to higher percentage of unprotected connections before reprovisioning. If resources are available to provision a new working lightpath (and a new backup lightpath if necessary) for each failed connection, then unnecessary reprovisioning of a large number of connections can be avoided. Therefore, under the premise that only new working connections are provisioned, resource sharability may have minimal impact on the performance of the reprovisioning algorithm. Notice, however, that this scheme does not neglect completely the effect of resource sharability. If the

Algorithm 3 Pseudo code of Scheme II in Reprovisioning

```
1: Each demand whose working path, $w_i$, is affected by the first failure is rerouted to its backup route, $b_i$, and resources along $w_i$ are released.
2: For each failed demand, find $w_i^{new}$ with enough capacity that is link-disjoint with $b_i$ and the primary routes of all demands sharing protection capacity with $w_i^{new}$. If successful: reserve the working capacity along $w_i^{new}$, and revert the traffic into it from $w_i^{new}$.
3: Otherwise, find $w_i^{new}$ with enough capacity that is link-disjoint with $b_i$ and find new protection capacity for $b_i^{new}$. If successful revert traffic to $w_i^{new}$, otherwise, compute new pair $(w_i^{new}, b_i^{new})$ and reserve corresponding capacity; if successful revert traffic to $w_i^{new}$.
4: For each failed demand that can not be reprovisioned in step 2 and 3, identify the list of all unprotected connections.
5: For each unprotected demand, release protection capacity that is already reserved and no longer useable. Repeat until all demands are processed.
6: Compute a link-disjoint route with the working path of each unprotected demand and allocate protection capacity if available.
7: Reserve capacity and repeat step 6 until all unprotected connections are processed or no reprovisioning is possible.
```

algorithm does not succeed in provisioning (some) new working connections for failed demands, then as in scheme I, the unprotected connections (those in categories 1 and 3) will be identified and reprovisioned. Accordingly, under this situation, the sharability level will have an impact on increasing the number of connections to be reprovisioned under this new scheme.

Two observations are in order here. First, resources along the routes of failed working connections are released (i.e., stub release) after the recovery from a failure which makes reversion to original paths impossible after the repair of the failed link. Although the issue of stub release is outside the scope of this paper, however, we note that leaving resources along the routes of failed connections unreleased (no stub release case) could degrade the immediate performance and increase the blocking rates of new arriving connections. Moreover, leaving resources reserved but unused may also increase the vulnerability of the network to new failures as searches for spare capacity to protect the exposed (unprotected) connections against a potential subsequent fault could fail due to unavailable resources. On the other hand, one drawback of stub release is that when a link is repaired, some of the connections may be routed through longer routes, which could result in semi-optimal operation of the network. Since spare capacity reconfiguration is done immediately after the occurrence of a fault, an operator that is concerned about the vulnerability of its network may want to exploit all available resources to reduce the risk for new failures. Therefore, we assume stub release in our study and for fair comparison stub release is implemented for both reconfiguration schemes.

Second, the drawback of Scheme II is manifested in the new service hit resulting from the reversion of affected traffic from active protection paths to new working paths. It is worth mentioning that this new traffic switch happens only upon the successful setup of these new working connections and hence may not necessarily result in extended recovery times. Note that in order to reduce the risk of the second service hit in a network with multiple classes of protection [54], an operator may choose not to allow higher class traffic to switch to new routes. Ultimately, it is a tradeoff between better reprovisioning performance (hence, higher robustness) and higher quality of service (in case second service hit may be considered as severely degrading the service of the network).

4.5 Centralized and Distributed Reprovisioning

The performance of reconfiguration strongly depends on the implementation of the underlying algorithm. An algorithm typically can either have a centralized or a distributed implementation [68]. Under a centralized implementation, a central network management system (NMS) holds the global information of network resources, such as network topology, link states, wavelength usage on each link, sharability information for protection resources, etc., and the corresponding steps of the particular algorithm (e.g., scheme I or II) are executed at this central controller. Here, upon the occurrence of a failure, the network will take the responsibility of recovering the failed connections through a standard signaling recovery protocol [47] and the central controller is informed through an alarm message to initiate the reprovisioning procedure.

Upon receiving an alarm, the central controller identifies the list of unprotected connections (if scheme I is deployed). For every unprotected connection in the list, a new protection path with available capacity is determined. The controller then configures resources by notifying each node along the route. After the controller receives acknowledgment from each node, it will send a message notifying the source node of the appropriate changes to its protection path. Similarly, when scheme II is used, the controller sequentially executes the steps in Algorithm 3 in order to avoid contention for capacity, which may lead to increasing the number of unprotected connections in the network and therefore increasing the vulnerability to a subsequent failure.

Alternatively, under distributed implementation of scheme I, the source node of each unprotected demand is responsible for reprovisioning new protection capacity for its connection. We deploy here a simple distributed provisioning approach with forward reservation [69], whereby the source node of one unprotected demand computes a new path and/or a new protection wavelength. Subsequently the node sends a control message containing the new selected wavelength to reserve resources along the entire path. If at least one node along the route is not successful in reserving the selected wavelength, the reservation fails and the connection is deemed unprotected. Here, unlike the centralized scheme where all connections are reprovisioned sequentially, all unprotected connections attempt to reserve protection capacity simultaneously and therefore contentions [11, 69, 68] may likely occur among connections requesting the reservation of the same resource. Clearly, a connection failing to find new protection capacity will be left unprotected and ultimately increases the network vulnerability to a subsequent failure. Further, if the number of unprotected connections simultaneously attempting to reprovision new protection capacity is quite large, contention over resources is more likely to increase. Therefore, to achieve a better network restorability, the effect of contentions will have to be reduced. The authors of [70] noted this problem of mass redial following a link failure to restore traffic; they argued that such problem arises only in an uncoordinated distributed framework (such a framework they refer to as distributed with blind reattempts). As a result, a blind scheme could yield extended restoration times and degraded overall recovery success. Hence, a form of overall coordination of the multiple simultaneous reprovisioning attempts is required. A scheduling method to coordinate the restoration process of multiple simultaneous connections is presented in [38].

Similarly under scheme II, when the failed connection is restored to its protection path, the source node attempts to provision a new working path (w_i^{new}), or if fails, to provision a new pair (w_i^{new}, b_i^{new}) by using forward distributed reservation. If this step fails, then the source node of this current failed connection will identify the list of unprotected demands (resulting from occupying the shared protection capacity) and subsequently inform their source nodes to reprovision new capacity to protect their connections (step 4 and after).

As before, contentions are likely to occur among multiple connections simultaneously attempting to provision new capacity (i.e., wavelength resources); therefore resulting in an increase in the number of unprotected connections after reprovisioning.

One of the advantages scheme II possesses over scheme I is that the number of connections to be reprovisioned is potentially much smaller; therefore making the impact of contentions on network restorability less severe. To mitigate the impacts contentions may have on the network restorability, some form of reconfiguration coordination among contending connections is required as mentioned previously. One simple method that is used in this work, however, is to allow unprotected connections attempting to reprovision and failing to succeed (due to contention) to *reattempt* [71] a new provisioning after selecting a different wavelength if possible. The advantage of reattempting is that blocking due to contentions may be reduced; however, the drawbacks are increased network reprovisioning times. Later we will see that reprovisioning retries strongly reduce the impact of contentions, and accordingly improves the network robustness.

4.6 Simulation Results

We study the performance of lightpath reprovisioning in nation wide network (Figure 4.4) consisting of 24 nodes and 43 bi-directional links. Requests are uniformly distributed between all source-destination pairs and arrive at each node via a Poisson process with a mean arrival rate of λ arrivals/ms. Meanwhile, the connection-holding time is exponentially distributed with mean $1/\mu$ m and the number of wavelengths per link is W=64[2]. A

[2] This flat capacity networks is artificial and is unrealistic of real networks. It is only used as a suitable test case for research purposes.

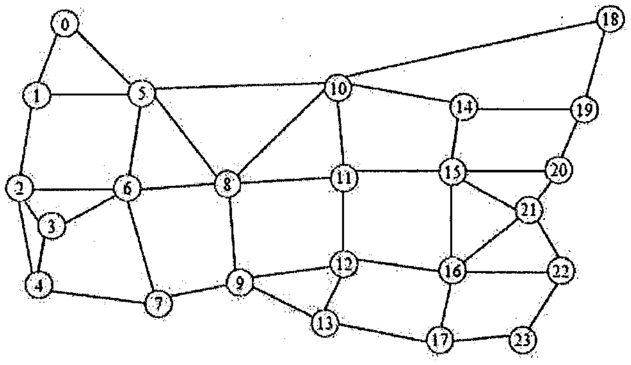


Figure 4.4: Sample Network Topology

standard disjoint shortest path algorithm is used for computing working and backup routes for each demand and a random wavelength assignment scheme is adopted [11]. Wavelength resources are pre-assigned on protection routes, however cross-connect switches are not configured at the time a connection is setup. Every node maintains a network state database reflecting the availability and sharability of wavelengths on its outgoing links. The capacity reprovisioning algorithm is assumed to run after the occurrence of each failure in order to protect the network and the carried connections from a potential next failure. In our simulation experiment, we assume that a failure occurs after a large number of connections have been admitted into the network. To measure the effectiveness of the reconfiguration algorithms, we take down randomly a second link upon completion of the reprovisioning process and measure the restorability of the connections in the network. Our results are averaged over all possible double link failures. The simulation operating points (i.e., blocking rates) vary between 0 at a load of 100 Erlangs and 0.2748 at 1000 Erlangs.

Table 4.1 summarizes the performance of reprovisioning under centralized implementation. We compare the conventional scheme (Scheme I) versus the proposed scheme

(Scheme II) in terms of total number of unprotected demands before reprovisioning (U_i), total number of demands to be re-provisioned (R_i), total number of successfully re-provisioned demands (SR_i), and the total number of unprotected demands after reprovisioning (UA_i, these are the connections that are vulnerable to new failures). We simulate the failure of a link and calculate the number of unprotected demands upon the failure (i.e., before reprovisioning); note that this number is the same for both schemes and it is equal to the number of connections to be re-provisioned in Scheme I (i.e., $U_1 = U_2 = R_1$). For Scheme II, the number of unprotected connections after reprovisioning and the number of successfully re-provisioned connections are measured to determine the total number of reprovisioned connections (i.e., $R_2 = UA_2 + SR_2$).

Loads	R_1	UA_1	SR_1	R_2	UA_2	SR_2
100	37	0	37	16	0	16
200	48	0	48	23	0	23
300	72	3	69	34	0	34
400	104	3	101	47	0	47
500	146	9	137	63	1	62
600	152	13	139	74	3	71
700	153	20	133	94	10	84
800	167	19	148	97	7	90
900	171	27	144	97	9	86
1000	177	34	143	113	20	93

Table 4.1: Scheme I vs. Scheme II - Centralized Reprovisioning

For example, in Table 4.1, shows that when the load is 500 Erlangs (i.e., blocking rate of 0.0182), the total number of unprotected connections resulting from the recovery upon first failure is 146. Under scheme I, a total number of 146 connections are reprovisioned and 9 connections are left unprotected (9:146). This shows that reprovisioning dramatically reduces the network vulnerability by protecting demands that are exposed to future failures.

Loads	R_1	UA_1	SR_1	R_2	UA_2	SR_2
100	37	9	28	17	1	16
200	48	17	31	24	1	23
300	72	30	42	44	7	37
400	104	48	56	55	9	46
500	146	71	75	91	14	77
600	152	83	69	100	25	75
700	153	83	70	120	29	91
800	167	83	84	113	21	92
900	171	90	81	135	36	99
1000	177	94	83	148	33	115

Table 4.2: Scheme I vs. Scheme II - Distributed Reprovisioning

On the other hand, scheme II shows that although 146 connections are unprotected, only 63 connections need to be reprovisioned and 1 connection is left unprotected out of 146 (1:146). Further, when the load increases, e.g. 1000 Erlangs, the gains of scheme I and II are 34:177 and 20:177 accordingly with only a total of 113 connections reprovisioned under scheme II. Clearly, the benefits of reprovisioning are evident. Our findings are in two aspects: (1) the number of unprotected connections in the network after reprovisioning is much lower in Scheme II than Scheme I. This indicates a better network restorability and less vulnerability to another failure; (2) the total number of connections that require reprovisioning upon a failure is lower in Scheme II. This yields a clear advantage as it can substantially lighten network management overheads and reduce contentions amongst simultaneously protection re-routing/reservation attempts. Overall, the results show that the proposed Scheme II performs less reprovisioning and yet achieves better protection.

The performance results of reprovisioning under distributed implementation are shown in Table 4.2. Similarly, the results show the advantages of network reprovisioning in reducing the total number of unprotected demands in the network after the first failure. However, it is important to notice that distributed reprovisioning protects fewer connections than the centralized scheme. This is mainly due to the fact that in a distributed environment, connections contend among each other to reserve protection capacity. For example, when the load is 1000 Erlangs, 94 connections (Table 4.2) are left unprotected after distributed reprovisioning (using scheme I) whereas only 34 connections (Table 4.1) are unprotected if reprovisioning is centralized. This therefore will adversely affect the network robustness in advance of a second failure.

Two observations are in order here. We first notice that the total number of reprovisioned connections for scheme II under distributed implementation is larger than that under centralized implementation (e.g., 148 vs. 113 at 1000 Erlangs). Unlike centralized reprovisioning, under distributed implementation our experiments showed that more connections will not be successful by using step 2 and 3 in Algorithm 3 (mainly due to contentions while simultaneously attempting reservation) and as a result, a lager number of connections will be reprovisioned by using step 4—7, therefore resulting in a larger number of reprovisioned connections. Moreover, note that the impact of contentions during reprovisioning is more severe on scheme I than scheme II. The justification for this is explained by the fact that under scheme I, all identified unprotected connections are reprovisioned simultaneously. Clearly the larger the number of connections reprovisioned simultaneously, the stronger is the impact of contentions. Alternatively under scheme II, the algorithm starts by reprovisioning only the directly affected connections (i.e., step 2 and 3 in Algorithm 3) and then it resorts to step 4—7 if necessary. Now, since the total number of failed connections is much smaller than the number of total unprotected connections, contentions will have a lower effect.

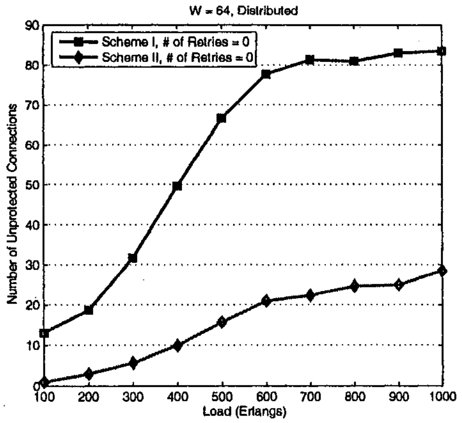


Figure 4.5: Number of unprotected connections due to contentions

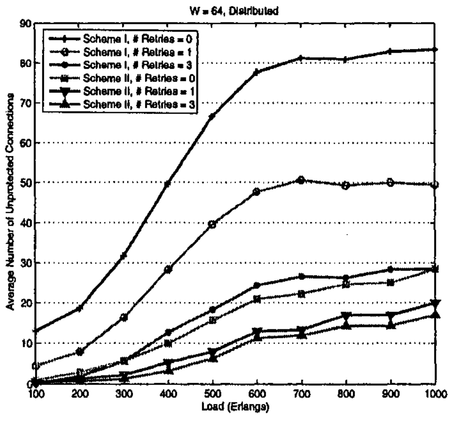


Figure 4.6: Number of unprotected connections due to contentions with retries

Overall, under distributed reprovisioning, a demand may fail to protect its connection for two reasons: (1) due to unavailable resources or (2) due to contentions with other connections. We measured the impact of contentions on increasing the number of unprotected connections in Figure 4.5. Clearly, most of unprotected connections fail to find protection capacity (wavelength) due to contentions while attempting to reserve resources. Also, figure 4.5 shows the strong impact contentions have on scheme I.

To minimize the impact of contentions, we propose that a connection that is being blocked due to only contention be allowed to select a new wavelength from the set of available wavelengths for this connection and retries its reservation. Figure 4.6 shows the benefits of this retry scheme in reducing the number of unprotected connections. This “reselect and retry” scheme improves the chance of future successful reprovisioning of connections blocked due to contentions. The figure shows that after one retry, the number of unprotected connections under scheme I reduces from 84 connections to 50 connections at 1000 Erlangs and to only 28 connections after 3 successive retries with much smaller (almost negligible) improvement after further retries. Similarly scheme II also benefits from "reselect and retry". However, scheme I benefits more than scheme II from retrying since larger number of blocked unprotected connections can retry until eventually blocking is due to only failure in finding available resources. The disadvantage of retrying, however, is the increase in the overall reprovisioning time. Our simulations showed that the total network reprovisioning time is kept well under 1 second when the total number of retries is 3.

Next we study the impact of resource sharability by measuring the percentage of unprotected connections in the network before and after reprovisioning. Figure 4.7 shows the performance of the two reprovisioning algorithms in a wavelength convertible (Figure 4.7(a)) and continuous (Figure 4.7(b)) network. As the level of sharability (or sharability index, SI[3]) of protection resources increases, the figure shows that the percentage of unprotected connections in the network before reprovisioning increases because connections are packed together and more connections are admitted to the network. This result is as well confirmed in a previous study by [62]. Clearly, capacity reprovisioning (under both schemes) improves the network performance by substantially reducing the percentage of unprotected connections (e.g., a decrease from 42% to 6% at higher SI in a wavelength convertible network using scheme I) and therefore making the network less vulnerable to subsequent failures.

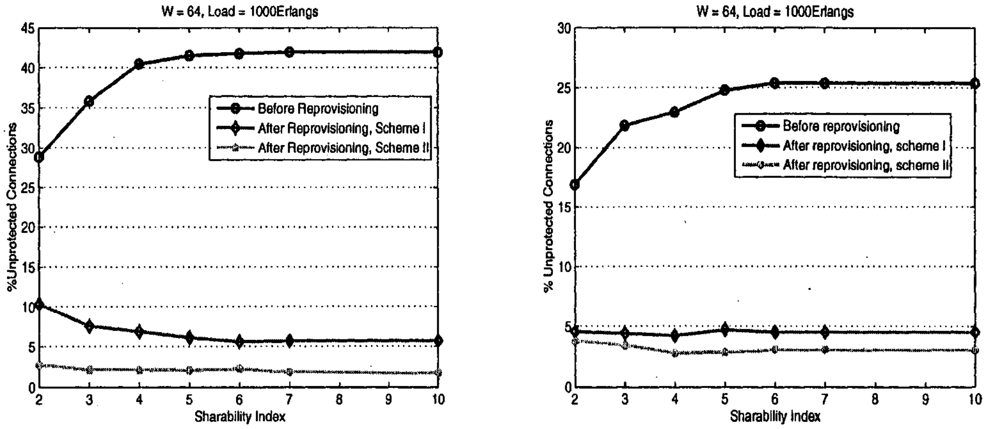
One interesting finding, shown in Figure 4.7(a), is that of the impact of SI on the reprovisioning gain. Namely, the lower percentage of unprotected connections before reprovisioning at lower SI does not necessarily mean a good reprovisioning performance (i.e., a

[3] The SI here varies between 2 and 10 where a SI = 1 corresponds to the dedicated protection case.

lower percentage of unprotected connections). The figure shows that as the sharability index increases, the percentage of unprotected connections after reprovisioning decreases for scheme I (10%-6%) while it remains almost constant for scheme II (~3%) with better performance than that of scheme I. The reason is that a lower SI will limit the flexibility of the reprovisioning algorithm in finding and judiciously allocating protection resources among unprotected demands. On the other hand, a higher SI will allow the network to accommodate more unprotected demands during reprovisioning by sharing the limited available resources. Therefore, the figure shows a larger performance gain at higher SI (~36% (42%-6%)) than at lower SI (20% (30%-10%)). Alternatively, scheme II shows a fixed percentage of unprotected connections at different sharability indices. The reason for that is due to the fact that scheme II gives preference to provisioning new working capacity (which is not impacted by SI) for failed demands in order to avoid reprovisioning a larger number of protection connections. Note that the reprovisioning gain is improved in Scheme II (~40% at higher SI vs.26% at lower SI) since more connections are admitted to the network at higher SI. Similar results are shown in Figure 4.7(b), except that the percentage of unprotected connections is smaller since in a wavelength continuous network, fewer connections are admitted to the network due to the wavelength continuity constraint.

A major metric of comparison we also use to evaluate the benefits of capacity reconfiguration is the network robustness. Here, robustness is defined as the capability of the network to maintain high restorability[4] of its connections (e.g., $\leq 95\%$) when two links are randomly taken down (one after the other). We measure the robustness before and after

[4] The restorability of a double failure (i, j), $R(i, j)$, is defined as the portion of all working paths $w_i + w_j$ on links i and j that are simultaneously affected and survive the failures [54, 55].



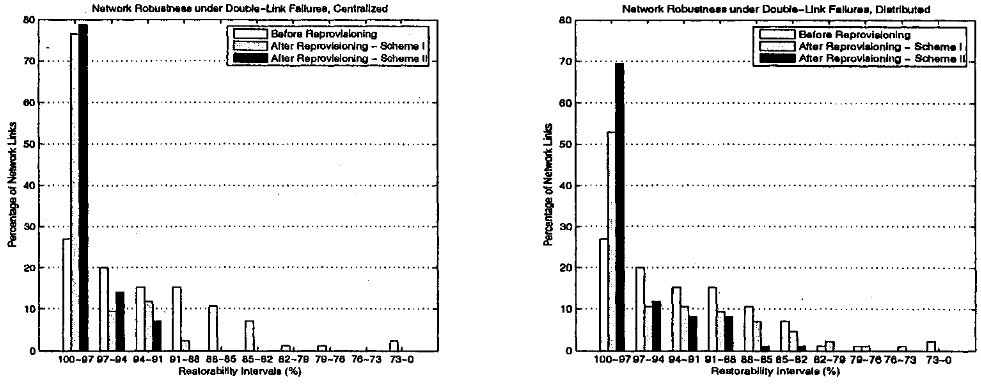
(a) Wavelength Convertible Network

(b) Wavelength Continuous Network

Figure 4.7: Impact of Resource Sharability on Reprovisioning

reprovisioning under unlimited resource sharability. The robustness is measured by first taking a link down and then measuring the restorability of the connections when another link fails from the remaining links. We then measure the percentage of links that result in a particular restorability value. This experiment is repeated for all links in the network and then we average all the results. Hence the larger the fraction of network links that yield higher connection restorability, the better is the overall robustness. In other words, given equal failure probability on all links, if dual failure restorability is kept at a desirable level for the majority of these links, then the network is said to be more robust.

We evaluate the network robustness of schemes I and II by the comparison of the double failure restorability before and after reprovisioning. Figure 4.8 shows the distribution of the number of links (percentage) with regard to network restorability intervals. We use 10 different intervals for the network restorability ranging from 0 to 100%. Namely, one large interval is chosen to cover a relatively low restorability range 0-73% and the remaining



(a) Centralized Implementation

(b) Distributed Implementation

Figure 4.8: Network Robustness at 1000 Erlangs

intervals are chosen in increments of 3% to cover higher ranges above 73%. Namely, in Figure 4.8(a), the figure shows the robustness of the network as the probability of having the restorability (R) within a certain interval. When the network does not use reprovisioning upon a failure, the 94% restorability is defined as Pr(R $\geq$ 94%) = Pr(R $\in$ [97% – 100%]) + Pr(R $\in$ [94% – 97%) = 0.47 (i.e., only 47% of the network links yields restorability above 94% after first recovery). After reprovisioning using Scheme I, this value increases to Pr(R $\geq$ 94%)=0.85 and even further to Pr(R $\geq$ 94%)=0.95 using Scheme II. The results show that the robustness improves substantially after reprovisioning; moreover Scheme II achieves significantly better robustness since the number of unprotected connections after reprovisioning here is much smaller than that under Scheme I.

Next, we present an evaluation of the reprovisioning technique under distributed implementation with no retries. As we mentioned in section 4.5, a major limitation under distributed implementation is the reduction in the number of successfully reprovisioned connections after the first failure due to contentions. Figure 4.8(b) shows the network robustness; Pr(R $\geq$ 94%) = 0.82 under scheme II whereas Pr(R $\geq$ 94%) = 0.63 under scheme I. This finding is explained by Table 4.2 since the number of unprotected connections after reprovisioning in scheme II is much smaller than that under scheme I (e.g., 34 vs. 94 at 1000Erlangs). Again, the results show a much better network robustness of scheme II over scheme I under distributed implementation.

Finally, we study the benefits of "reselect and retry" on improving the network robustness by reducing the impact of contentions when capacity reprovisioning is distributed. Figure 4.9 presents the percentage of links that achieve restorability above 97%. Clearly the figure shows that after only one "retry" attempt using reprovisioning scheme II, 75% of the network links can achieve restorability above 97%; this result is considerably good given that under centralized implementation of the same scheme, 78% of the network links can achieve restorability above 97%. On the other hand, scheme I benefits more of reservation retries since the performance is monotonically increasing (see Figure 4.9). However the performance is always inferior to that of scheme II, and moreover it is worth mentioning that when the number of retires increases, the overall reprovisioning time becomes longer. Note that Figure 4.9 also shows the network robustness before reprovisioning and that under centralized reprovisioning for comparison reasons.

4.7 Conclusion

We studied the problem of improving restorability in shared optical mesh networks for dual, near-simultaneous failure events in Chapter 4. A novel capacity reconfiguration scheme for

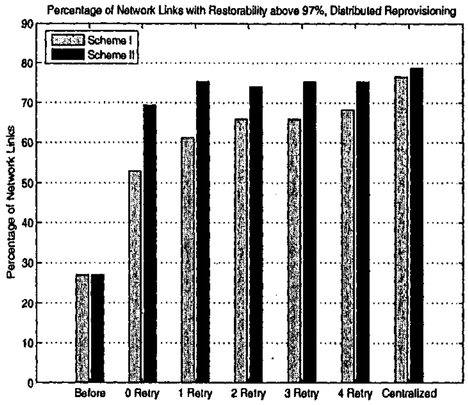


Figure 4.9: Network Robustness at 1000 Erlangs - with retries

shared backup path protection is introduced in order to reduce the number of unprotected connections after the first failure and in advance of a second failure. We showed that the new scheme re-provisions fewer connections and protects more demands. We discussed the implementations of reprovisioning under both centralized and distributed control and found that under simple distributed implementations, the robustness of the network degrades due to excessive contentions that occur when simultaneous connections attempt to reconfigure their capacity. We also showed that the proposed reprovisioning scheme performs better than the conventional scheme under distributed implementation since the number of resulting unprotected demands is lower and accordingly the contentions during reconfigurations is less. Finally, we presented a simple method to reduce the impact of contentions by allowing unsuccessful connections to reattempt reprovisioning and we validated our work through extensive simulations experiments. One drawback, however, for the proposed algorithm is manifested in the possible second service hit for recovered connections in order to reduce the number of exposed demands. Therefore, an operator must weigh the gain achieved from spare capacity reconfiguration (in terms of improved robustness again dual failures) vs. the slight reduction in the quality of service provided by the network.

Chapter 5

Multiple-Link Failures Survivability in Optical Networks with Traffic Grooming Capability

5.1 Introduction

As we mention in Chapter 4, recent research has focused on improving the service availability of these networks against multiple simultaneous failures either through preplanned redundant capacity or through capacity reprovisioning or further using p-cycle reconfiguration in mesh networks. Most, if not all, of these efforts have assumed that every user demands a bandwidth equals to the full wavelength capacity. Currently, the transmission rate of a wavelength channel is STS-192 (10Gbps) and expected to grow to STS-768 (40Gbps) in the near future. Bandwidth requirement of a typical connection request varies, however, from full wavelength capacity to as low as STS-1 or lower. Hence, it is necessary to efficiently pack these lower speed demands (or connections) onto high capacity light channels (also known as lightpaths) in order to better utilize the network resources. This problem has emerged lately and is known as the traffic grooming problem [72, 73, 74, 75, 76, 77]. Traffic grooming refers to the problem of efficiently packing low-speed connections onto high-capacity lightpaths in order to better utilize the network resources and has been studied extensively over the past years both for SONET/WDM ring networks [76, 77] as well as in optical mesh networks [72, 73, 74, 75].

Now, how to efficiently groom such low-speed connections while satisfying their protection requirements is best known as survivable traffic grooming (STG) problem and presently is attracting some considerable research efforts [78, 79]. The authors of [78] have proposed different frameworks for protecting low-speed connections against single link failures in optical mesh networks and have shown that providing collective protection of connections at lightpath level (PAL) achieves better performance than protecting at the connection level (PAC) while it also requires a smaller number of grooming ports. To make connections survivable under various failures, such as fiber cut and duct cut, the authors of [79] studied the static STG problem under the general shared risk link group diverse routing constraints where protection is provided at the lightpath level. They formulated the problem as an integer linear program and proposed several heuristics.

In this chapter we revisit the problem of survivable traffic grooming in mesh networks and we study the survivability of connections against multiple concurrent failures where concurrent implies that the new failure occurs before the previous failure has been repaired.

We focus on mesh networks that are only designed to withstand all single link failures either through lightpath level protection or through connection level protection with shared backup resources. To combat the effect of multiple failures, and hence improve the service availability, we propose to use capacity reprovisioning, as discussed in Chapter 4. We present lightpath level and connection level reprovisioning as complementary approaches for survivable traffic grooming to achieve better service robustness against multiple failures. Note that when connections are protected at the lightpath level, the process of reprovisioning takes place at that level (thereafter referred to as lightpath level reprovisioning, LLR); in other words, only the lightpaths that become unprotected/vulnerable need to be reprovisioned. Alternatively, if the connections are protected at the connection level, reprovisioning takes place at connection level (thereafter referred to as connection level reprovisioning, CLR). We compare the performance of these two schemes and present some results on the robustness of the network under both frameworks. Our results show that connection level reprovisioning substantially outperforms the lightpath level reprovisioning.

The rest of the chapter is organized as follows. Section 5.2 presents an overview of the survivable traffic grooming problem and we present some simple heuristics and compare their performance. Section 5.3 presents a detailed study of the reprovisioning approaches and we quantify their performances in section 5.4. Finally, we conclude in section 5.5.

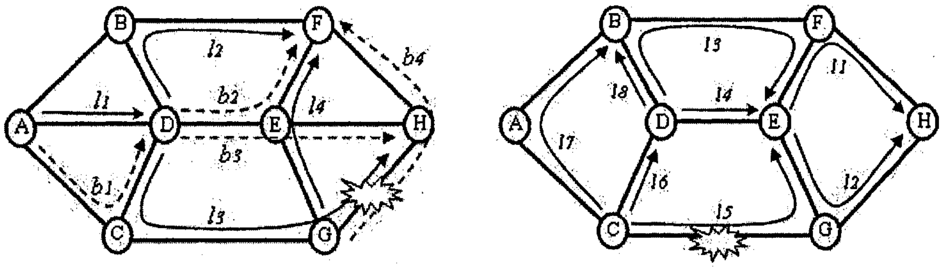
5.2 Survivable Traffic Grooming

5.2.1 Background

Grooming connections while still satisfying their protection requirements is known as survivable traffic grooming, STG [78, 79]. Different schemes have been proposed for protecting connections, namely protection at lightpath level (PAL), mixed and separate protection at connection level (MPAC and SPAC).

Under PAL, a connection is typically routed through a sequence of protected lightpaths (*p-lightpaths*) where a *p-lightpath* is a pair of working and link-disjoint backup lightpaths. A working lightpath consumes one grooming add port and one drop port and wavelengths along the route of a lightpath are reserved and configured. Resources for a protection lightpath, on the other hand, are only reserved and they are setup after the failure. Hence, a protection lightpath does not consume any grooming add/drop ports. Normally, a demand is routed over a multihop route (sequence of *p-lightpaths*) if there is no direct lightpath with enough capacity connecting the source and the destination of the demand. In case of a failure along the working lightpath, the carried traffic (i.e., the set of connections routed through this lightpath) is restored onto the protection lightpath; only the end (and intermediate) nodes of the lightpath are aware of the switching and the end nodes of the failed connections are oblivious to this protection switching.

Figure 5.1(a) shows an illustrative example of 4 *p-lightpaths*. A demand, d_1, between nodes A and F can be routed through lightpaths (l_1-l_2) where l_1 and l_2 are protected by b_1 and b_2 respectively. Note that under PAL two *p-lightpaths* can share wavelengths along their protection lightpaths if their corresponding working lightpaths are link disjoint. For



(a) PAL

(b) PAC

Figure 5.1: Illustrative examples of STG

example, l_2 and l_3 can share the same protection wavelength along link (D-E) since they are link disjoints.

Alternatively, PAC provides end to end protection at the connection level and has two variants (SPAC and MPAC) that differ in the way connections are protected [78]. In SPAC, a connection is routed via a link-disjoint working and backup routes. The working traverses a sequence of lightpaths and the backup traverses a sequence of wavelength links, where each wavelength link consumes a pair of grooming ports, add and drop, at each end of the link. Under MPAC, a demand is routed via link-disjoint working and backup paths each traversing a sequence of lightpaths. Each lightpath consumes a pair of grooming ports, one add port at the source and one drop port at the destination. Every lightpath traversed by a working connection reserves a fixed amount of bandwidth to carry the demand traffic. A lightpath that is traversed by a backup connection correspondingly reserves a fixed amount of its capacity to protect against the failure of the working connection. In this thesis, we will only use MPAC and we use the term PAC to refer to this grooming policy.

A working connection fails when any of the lightpaths that it traverses fails. Upon the failure, the source node of the failed demand switches traffic from the working connection into its corresponding backup. ***Bandwidth sharing*** is achieved under MPAC when two demands have their corresponding working connections physically end-to-end link-disjoint and their backup connections traverse the same lightpath(s). Figure 5.1(b) shows an example of PAC grooming. The figure shows a set of existing lightpaths; when a new demand d_1 arrives (e.g., between nodes C and H and demanding a bandwidth of STS-12), it is routed through l_5 and l_2 and is protected by lightpaths l_6, l_4 and l_1. A demand, d_2, between nodes D and E and bandwidth 2×STS-12 can be routed through l_3 and protected by l_4. Here, d_1 and d_2 are both end-to-end link-disjoint and both share the same lightpath l_4, hence they can both share the protection bandwidth reserved along l_4 and the new protection bandwidth reserved along l_4 becomes 2×STS-12. When a link fails, the lightpaths routed through that link will fail and hence the connections routed through every failed lightpath will also fail. As mentioned before, those failed connections will be restored by their corresponding source nodes.

5.2.2 Comparison between PAL and PAC

For a comprehensive quantitative and qualitative comparison between PAC and PAL frameworks, we refer the reader to the work of [78]. These schemes differ in two main characteristics, namely the routing and the backup bandwidth sharing. In terms of routing, PAL provides end to end protection at the lightpath level whereas PAC provides end to end protection at the connection level. In PAL, after the failure of a link, the end nodes of a failed lightpath configures the protection lightpath and switch the traffic into it and the end nodes of the connections are unaware of this process. Alternatively, in PAC the end nodes of the failed connections configure their backup paths and restore the traffic. Under PAL, only working and protection paths of a *p-lightpath* must be link disjoint and the working and protection path of a demand need not be end to end link-disjoint. In PAC, however, the working and backup routes of a demand must be end to end link disjoint. Hence, when a demand spans multiple lightpaths, it becomes difficult to find an end to end link disjoint protection path to protect the connection.

With respect to backup sharing, protection wavelength links are the resources that can be shared in PAL. Namely, two *p-lightpath*s can share the same protection wavelength link if their working lightpaths are link disjoint and their protection lightpaths traverse through that same protection wavelength. Hence, all working connections traversing these *p-lightpath*s are said to be sharing that protection wavelength link. However, under PAC (i.e., MPAC) the sharing unit is a lightpath (or the backup bandwidth reserved in a lightpath). Hence, two demands (d_1 and d_2) under PAC can share protection bandwidth in a lightpath l if (1) their corresponding working connections are end to end diversely routed and (2) their protection connections traverse lightpath l. Then the backup bandwidth required on lightpath l is $\max(bw_1, bw_2)$ where bw_1 and bw_2 are the bandwidth requirements of demands d_1 and d_2 respectively. Clearly, since a lightpath may traverse multiple physical links and a connection is routed through multiple lightpaths, it is less likely that conditions (1) and (2) are together satisfied and hence bandwidth sharing is hard to achieve. All these reasons make the PAC algorithm less attractive than PAL; the authors of [78] have evaluated using simulations the performance differences between PAL, SPAC and MPAC. Overall results showed that PAL achieves best performance when the number of grooming ports is either small or moderate.

Moreover, as mentioned earlier, under PAL only the working lightpath consumes add/drop grooming ports whereas under PAC every lightpath consumes add/drop grooming ports (note in SPAC every wavelength link along the protection route consumes one pair of add/drop ports). PAC, on the other hand, allows both working and protection connections of different demands to be routed through the same lightpath, a flexibility that does not exist under PAL.

5.2.3 STG Grooming Heuristics

Clearly, while PAL trades the flexibility in grooming for the freedom of backup sharing, PAC allows working and backup connections of different demands to be groomed on the same lightpath. However, one major drawback for PAC is the difficulty in sharing backup bandwidth among the demands. This drawback is further exacerbated as the physical hop count of lightpaths gets larger. So we propose some simple modifications to improve the efficiency of backup bandwidth sharing. We note first that if the physical hop count of every lightpath is limited to one, then in terms of backup sharing MPAC becomes similar to SPAC; further, since MPAC allows protection and working capacity to be groomed on the same lightpath, then this new MPAC-1 (1 means a lightpath is limited to one hop) achieves both high flexibility and better bandwidth sharing. However, the grooming capacity requirement could be excessive. Therefore, we propose to limit the length of a lightpath in order to improve the backup bandwidth sharing while still maintaining the flexibility of routing. This new version of PAC is referred to as PAC-HCL, where HCL refers to Hop Count Limit and HCL ≥ 1.

Next, we explain the STG heuristic; in response to a new connection request, PAC-HCL first computes two link disjoint paths from source to destination using *Dijkstra* Algorithm in the existing logical topology. Every lightpath along the working path must have enough bandwidth to carry the new demand. Along the protection path, bandwidth sharing is used. Every lightpath (l) reserves backup bandwidth (v_l) to protect all the connections whose protection paths traverse through this lightpath. Note, $v_l = max_{\forall e' \in E}\{v_l^{e'}\}$, where $v_l^{e'}$ is the amount of bandwidth reserved on lightpath l to protect against the failure of link e' ($0 \leq v_l^{e'} \leq OC-192$) and E is the set of all links in the network. When a new connection of bandwidth w is protected by l, the additional backup bandwidth reserved on l is α_l and is determined as follows: $\alpha_l = max\{v_l^{new} - v_l, 0\}$, where $v_l^{new} = max_{\forall e' \in E}\{v_l^{e',new}\}$, and $v_l^{e',new} = v_l^{e'} + w$ if the working connection of the new demand traverses through e', otherwise $v_l^{e',new} = v_l^{e'}$. In case there is not enough bandwidth to route and/or protect the demand on the logical layer[1], then a new lightpath(s) is setup on the physical layer for either the working or protection or both paths.

At the physical layer, the source node computes the shortest path route to the destination ($s - x_1 - x_2 - \ldots - x_n - d$). The source will select a node x_i from the shortest path that is HCL hops away and check whether there is a direct lightpath (already setup) with enough capacity (or with enough sharable capacity in case of a protection connection). If there is not, the source node checks for a node that is HCL-1 hops away from the source node and so on until a lightpath is found. If a node (x_i) is found, the same procedure as before is run again between x_i and the destination. Let x_t ($x_t = s$ if there is no direct lightpath between s and any node on its shortest path that is at most HCL hops away from s) be the

[1] A lightpath layer or logical layer is the set of lightpaths currently in the network.

node after which there is no outgoing lightpath(s) to any node along the shortest path to d. At this point, the algorithm tries to "setup" a new lightpath from x_t to a node that is HCL hops away. If that fails, then a node that is HCL-1 hops away is checked and so on until a lightpath is setup. If a lightpath is found, then the same procedure is repeated until a route is established all the way to the destination. At any step, if a lightpath could not be setup, the request is dropped and all allocated resources are released. The detailed steps are presented in Algorithm 3. With regards to PAL, the same algorithm as in [78] is implemented.

Algorithm 4 Pseudo code of the Provisioning in PAC-HCL

```
Input: A network represented as a directed graph $G = (V, E, \lambda, P)$, where $V$ is a set of
    nodes, $E$ is the set of unidirectional physical links, $\lambda$ specifies the number of wave-
    lengths on each link, and $P$ specifies the number of grooming ports at each node. The
    logical topology of this network is represented as a graph $G' = (E, L)$, where $L$ is the
    set of lightpaths.
Output: Link-disjoint working and backup paths, or NULL if fails.
1: Compute a pair of shortest and physically link-disjoint logical paths (i.e., a working
    path with enough bandwidth and a protection path with enough sharable bandwidth)
    from src to des; if successful, return the two paths, otherwise compute the shortest
    physical working path $wroute = (src - x_1 - x_2 - \ldots - des)$ and go to step 2.
2: The node $s$ ($s = wroute[0]$) will check whether there is an existing direct lightpath from
    $s$ to a node along the path which is HCL hops away and has enough capacity; if there
    is not, check for a lightpath with HCL-1 hops from $s$ and so on until a lightpath is
    found. If there is such a lightpath (assume its destination node is $x_t$), then $wroute \leftarrow$
    $(x_t - x_{t+1} - \ldots - des)$ and repeat step 2 on wroute. If a working path has been found
    to the destination, go to step 4, otherwise, go to step 3.
3: Let $x_t$ be the node after which there is no outgoing lightpath(s) to any node along
    wroute. Then, try to establish a new lightpath from $x_t$ to a node that is HCL hops away.
    If this fails, then a node that is HCL-1 hops away is checked and so on until a new
    lightpath is setup. If a lightpath is setup and assume its destination node is $x_{t'}$, then
    $wroute \leftarrow (x_{t'} - x_{t'+1} - \ldots - des)$ and go to step 3. If a working path has been found
    to the destination, go to step 4; otherwise, return NULL.
4: Eliminate the working path in graph G, find the shortest physical path broute from src
    to des; repeat the similar procedure in steps 3 and 4 on broute to provision the backup
    path. If successful, return working and backup paths; otherwise, release the resources
    reserved along the working path and return NULL.
```

5.3 Connection and Lightpath Level Reprovisioning

In Chapter 4, we have discussed the motivation and mechanism of Capacity Reprovisioning, which provides a mechanism by which one can find and allocate new protection capacities for these newly-unprotected lightpaths without a priori knowledge of the location of the second failure. In this section, we apply the idea of reprovisioning in STG problem. Namely, we consider low-speed connection requests and propose two frameworks of reprovisioning for improving their survivability against multiple failures. The first scheme is lightpath level reprovisioning (LLR) that relies on PAL and the second is connection level reprovisioning (CLR) which rather relies on PAC.

5.3.1 LLR

As mentioned before, under PAL a connection traverses a sequence of *p-lightpaths*. The working route of a connection traverses the sequence of working lightpaths and is protected by the sequence of corresponding protection lightpaths. Consider every link in the network to be associated with a conflict set to identify the sharing potential between protection lightpaths [78, 80]. The conflict set v_e for link e can be represented as an integer set, $\{v_e^{e'} \mid \forall e' \in E, 0 \leq v_e^{e'} \leq \lambda(e)\}$, where $v_e^{e'}$ is the number of working lightpaths that traverse link e' and are protected by link e, E is the set of all links in the network, and $\lambda(e)$ is the number of wavelengths per link e. Then, the number of protection wavelengths reserved on link e to protect against the failure of any other link in the network is given by $v_e^* = max_{\forall e' \in E}\{v_e^{e'}\}$.

When a link (e.g., f) fails, all lightpaths routed through that link also fails and accordingly all the connections carried by these lightpaths fail. The failed lightpaths are rerouted onto their corresponding protection lightpaths and consequently become unprotected and exposed to a new failure. For example, when link (G-H) in Figure 5.1(a) fails, then lightpath l_3 fails and is restored into its protection lightpath b_3. All connections routed through l_3 will fail and will be restored to b_3. Note that b_3 and the restored connections are all exposed to a new failure. Moreover, all the demands that were originally protected by link f have lost their protection resources and become unprotected. For example, all connections traversing l_4 now become unprotected.

Upon the recovery of the failed lightpaths to their protection routes, some backup wavelengths on a link, say e, may be activated if at least one of these protection lightpaths traverses through link e. Hence, the number of new available protection wavelengths on link e is $v_e^a = v_e^* - v_e^f$ and the number of protection wavelengths on link e required to protect against a future link failure in the network is $v_e^{new} = max_{\forall e' \in E-f}\{v_e^{e'}\}$. If $v_e^{new} > v_e^a$, then some of the existing lightpaths that were not directly affected by the failure are vulnerable to a new failure because link e does not have enough protection capacity. For example, before the failure, link (D-E) reserves only one wavelength ($v_{DE}^* = 1$) to protect l_2 and l_3. When l_3 is rerouted to b_3 after the failure, $v_{DE}^a = 0$ and $v_{DE}^{new} = 1$, hence l_2 is vulnerable to a new failure.

Let $e_1, e_2, \ldots, e_F$ be the set of links traversed by some active protection lightpaths after the first failure; then the set of all vulnerable lightpaths can be identified. A connection that traverses an unprotected *p-lightpath* is unprotected and similarly a connection that traverses a vulnerable *p-lightpath* is also vulnerable. In LLR, the resources along the failed lightpaths are released, and every unprotected lightpath that is identified is reprovisioned by computing and allocating new protection capacity for this lightpath. On the other hand, some of the vulnerable lightpaths need to be reprovisioned in order to reduce the vulnerability of the network to a second failure. When a vulnerable lightpath is reprovisioned, some other vulnerable lightpaths may become protected [60, 39] if they were originally contending with the reprovisioned lightpath for the same protection wavelength on a particular link. Hence, a vulnerable lightpath l becomes protected after the reprovisioning of another lightpath, when for every link (e) along the backup lightpath we have $v_e^a \geq max_{\forall e' \in E-f}\{v_e^{e'}\}$. Each time a new vulnerable lightpath is reprovisioned, the set of remaining vulnerable lightpaths is identified; this procedure continues until all vulnerable lightpaths are reprovisioned or no more reprovisioning is possible. Note that there are many policies [60] for selecting a vulnerable lightpath from the set, for simplicity we select a vulnerable lightpath randomly.

Algorithm 5 Pseudo code of LLR

```
1: Identify a set $L^u$ composed by unprotected lightpaths $(l_1^u, l_2^u, \ldots, l_m^u)$, then release unavailable resources along these unprotected lightpaths and reprovision them by allocating new protection wavelength(s) in the physical topology, as explained before. If an unprotected lightpath $l_i^u$ cannot be reprovisioned, move it to another set $L_{after}^u$.
2: Identify a set $L^v$ composed by vulnerable lightpaths $(l_1^v, l_2^v, \ldots, l_m^v)$. For each vulnerable lightpath $l_i$, reprovision it using the same method as in step 1. If not successful, move $l_i$ to another set $L_{after}^v$; otherwise, remove $l_i$ from $L^v$ and re-identify other vulnerable lightpaths in $L^v$ and $L_{after}^v$, then repeat step 2 until there are no vulnerable lightpaths or no more reprovisioning is possible.
```

5.3.2 CLR

CLR is used when connections are protected at the connection level. Recall that a connection traverses a sequence of lightpaths and is also protected by a sequence of link disjoint lightpaths. The backup sharing between two connections is at the lightpath level, see section 5.2. Let $A_l^{e'}$ be the set of all connections c_i $(i = 1, \ldots, M)$ each with bandwidth w_i

traversing physical link e' and protected by lightpath l. The bandwidth reserved on lightpath l to protect against the failure of link e' is $v_l^{e'} = \sum_{i=1}^{M}(w_i)$, $0 \leq v_l^{e'} \leq STS-192$. The total amount of backup bandwidth reserved on lightpath l to protect against the failure of any link e' in the network is $v_l^* = max_{\forall e' \in E}(v_l^{e'})$.

When a link f fails, all lightpaths traversing link f fail and accordingly all connections routed through these lightpaths will also fail. These connections will be rerouted onto their protection routes and become unprotected and hence exposed to new failures. For example, a connection c_1 of bandwidth 2×STS-1 between nodes C and H is routed through working path (l_5-l_2) and protected by (l_6-l_4-l_1), see Figure 5.1(b). When link (C-G) fails, the connection is restored to its end to end backup path and after recovery, the connection becomes exposed. Similarly, all connections that were originally protected by any lightpath traversing link f also become unprotected. For example, a connection (c_2) between nodes C and E whose working is (l_6, l_4) can be protected by l_5. Hence, when (C-G) fails, l_5 fails and the connection c_2 loses its protection bandwidth and becomes unprotected. Now, when a connection is restored into its protection route, the backup bandwidth reserved on any lightpath along the protection route for this connection is activated and can no longer be shared. Hence, a lightpath l will have $v_l^a = v_l^* - v_l^f$ available protection capacity to protect against a new failure. The protection capacity required, however, on lightpath l to protect against the future failure of any link is $v_l^{new} = max_{\forall e' \in E-f}\{v_l^{e'}\}$. For example, if a connection c_3 (4×STS-1) between nodes D and E (Figure 5.1(b)) has its working traversing l_3 and protected by l_4; this connection can share protection bandwidth along l_4 with connection c_1 since the working paths of these two connections are link disjoint. Hence, l_4 reserves max(2×STS-1,4×STS-1)=4×STS-1 to protect these two connections.

When link (C-G) fails, the available backup bandwidth on l_4 becomes $4\times$STS-1$-2\times$STS-1=$2\times$STS-1 which is not sufficient to protect c_3. Hence, c_3 becomes vulnerable to a new failure.

Let C_l be the total capacity of a lightpath, R_l be the residual capacity, A_l be the bandwidth used by working and *active* backup connections; hence, $R_l = C_l - A_l - v_l^a$. Let $\Delta = \{l_1, l_2, \ldots, l_L\}$ be the set of all lightpaths on which failed connections are rerouted. If for every $l_i (i = 1, \ldots, L)$, (1) $v_{l_i}^a \geq v_{l_i}^{new}$, then there will be no vulnerable connections in the network; or (2) $v_{l_i}^a + R_{l_i} \geq v_{l_i}^{new}$, then the lightpath l_i has enough available capacity (and should be reserved) that can protect against the failure of any link in the network. Hence, the backup capacity reserved along l_i becomes $v_{l_i}^{new} = max_{\forall e' \in E-f}\{v_{l_i}^{e'}\}$. In this case, only unprotected connections as mentioned earlier need to be reprovisioned. Alternatively, when (1) and (2) are not satisfied for at least one lightpath l_i, then some connections in the network are vulnerable to a new failure. In this case, the set of all vulnerable connections is identified (as in LLR); a vulnerable connection is reprovisioned by allocating new protection capacity on its backup route. The set Δ is updated and conditions (1) and (2) are checked again for vulnerable connections. The same procedure is repeated until there are no more vulnerable connections or no more reprovisioning is possible.

5.3.3 LLR vs. CLR

Both of these schemes rely on spare capacity reprovisioning after the first failure in order to improve the network survivability against a new failure. However, the two schemes present some critical differences.

Algorithm 6 Pseudo code of CLR

```
1: Identify a set $U$ of unprotected connections $C_1^U, C_2^U, \cdots, C_m^U$.
2: Reprovision each connection $C_i^U$. If not successful, move $C_i^U$ from $U$ to another set
   $U_{after}$ (which stores the unprotected connections after reprovisioning). Repeat step 2
   until $U$ is empty or no more reprovisioning is possible.
3: Initialize a set $V$ composed by vulnerable connections $C_1^V, C_2^V, \cdots, C_m^V$, which are iden-
   tified by the model described in section 5.3.2.
4: Reprovision connection $C_j^V$. If successful, remove $C_j^V$ from $V$; otherwise, move $C_j^V$ to
   a set $V_{after}$ and in both cases re-evaluate the remaining vulnerable connections in the
   network (i.e., two sets $V$ and $V_{after}$) for vulnerability; those that become protected are
   removed from the two sets. Repeat step 4 until $V$ is empty and exit. All connections in
   $V_{after}$ are vulnerable and exposed for future failures.
```

The first difference between the two schemes pertains to the granularity at which each scheme reprovisions protection bandwidth for its demands. In LLR, the end nodes of the demands are not aware of this reprovisioning process. The source node of a failed *p-lightpath* reconfigures new backup resources without the intervention of end nodes of the connections it carries. Therefore, when an unprotected or vulnerable lightpath is successfully reprovisioned, all demands traversing this lightpath becomes protected; conversely, if the lightpath could not be reprovisioned, then all the demands it carries are unprotected. Thus, LLR provides collective reprovisioning for low speed connections. On the other hand, in CLR connections are reprovisioned at a smaller granularity. Here, the number of connections that are unprotected or vulnerable is substantially much more than the number of unprotected or vulnerable lightpaths (although the number of unprotected or vulnerable connections in both cases may be the same). In CLR, the end node of every unprotected/vulnerable connection needs to reprovision new protection capacity; hence, the management overhead may be excessive.

A second difference between LLR and CLR pertains to the method of reprovisioning.

In LLR, all lightpaths are reprovisioned by setting up new protection resources on the physical layer. If resources are not available, then the reprovisioning fails and the lightpath remain unprotected. Alternatively, under CLR, unprotected connections are reprovisioned first on the logical layer; that is, the algorithm first attempts to allocate protection resources on already existing lightpaths. If this fails, then the physical layer is requested to setup new lightpaths. Hence, although the number of unprotected connections to be reprovisioned (under CLR) is much larger than the number of unprotected lightpaths (under LLR), CLR enjoys more flexibility for capacity reprovisioning. This is particularly advantageous when CLR is implemented with PAC-HCL since this latter has better flexibility and more efficient bandwidth sharing than PAC.

5.4 Performance Evaluation

This section presents quantitative comparisons between PAL, PAC and PAC-HCL (HCL=3 throughout the simulations) presented in section 5.2 and it also compares the performance of LLR and CLR presented in section 5.3. We simulate a dynamic network environment where connection requests are uniformly distributed between all source-destination pairs and their arrival process is Poisson. The connection holding time of each connection follows a negative exponential distribution. The network we simulated is nation wide network (Figure 4.4), and the number of wavelengths per link is 8. The capacity of each wavelength is STS-192; the number of the connection requests follows the distribution STS-1 : STS-3c : STS-12c : STS-48c = 12 : 5 : 2 : 1. The load (in Erlangs) is defined as the arrival rate of connection requests times average holding time times a connections average bandwidth

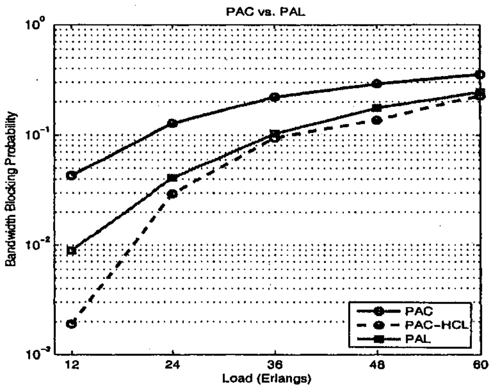


Figure 5.2: Bandwidth Blocking Probability

normalized in the unit of STS-192.

5.4.1 Provisioning Results

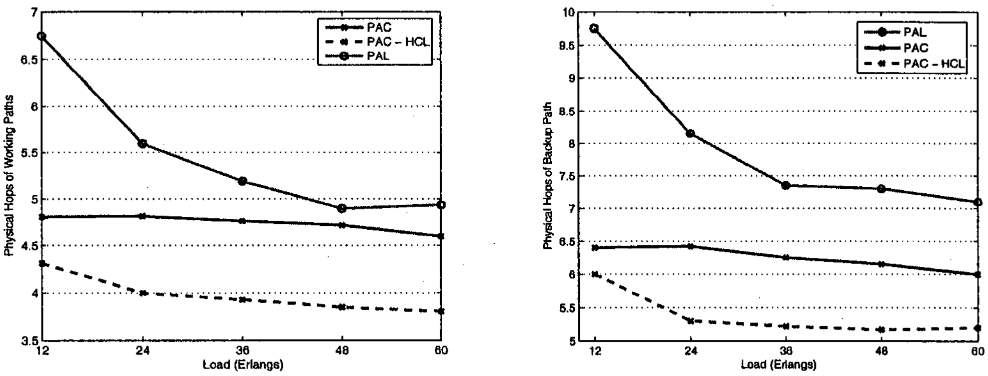
We compare PAL, PAC and PAC-HCL using the following metrics: Bandwidth Blocking Probability (BBP), length of working and backup paths, impact of grooming capacity, efficiency of backup sharing.

Figure 5.2 shows the BBP for the different grooming schemes; the number of grooming ports is 16 (16 add and 16 drop) per node. The BBP is defined as the amount of bandwidth blocked over the amount of bandwidth requested. The figure shows that PAL and PAC-HCL have comparable performance with PAC-HCL slightly outperforming PAL; while PAC on the other hand is exhibiting worse performance than the other schemes. The reasons are as follows. First, under PAC, a connection traverses a sequence of lightpaths and is protected by another physically disjoint sequence of lightpaths. When a lightpath traverses more hops (i.e., is longer), finding two sets of lightpaths that are end to end physically disjoint becomes more difficult (physical disjoint constraint). Second, sharing of backup bandwidth under PAC is end to end; that is, as mentioned in section II, difficult to achieve under PAC due to the physical disjoint constraint particularly when lightpaths may traverse more physical hops. Third, in PAC although the bandwidth on the existing lightpaths may be available for carrying new connections, the physical disjoint constraint prevents some of these connections from being routed on the logical topology and instead they are routed on the physical topology by setting up new lightpaths and therefore consuming new wavelengths and increasing the bandwidth blocking probability.

Alternatively, in PAL, a connection traversing *p-lightpath*s does not necessarily have to be end to end disjoint; only the working and protection lightpaths of a *p-lightpath* need to be. Moreover, backup sharing is much more relax than PAC[2]. PAC-HCL, on the other hand, outperforms PAL since it allows the grooming of protection and working bandwidths on the same lightpath. It also outperforms PAC since restricting the hop count of a lightpath yields better flexibility in finding disjoint routes on the logical topology and furthermore backup bandwidth sharing is better exploited.

Figure 5.3 shows the physical hop count for working and protection connections in all three schemes. We have two findings here. First, connections under PAL are routed through longer routes than the connections under PAC, PAC-HCL. Note that PAL allows this because "backup sharing" condition and "physically link-disjoint" condition are not end-to-end and rather they are only at the lightpath level. Secondly, as the load increases, physical hops of working/backup paths in all schemes decrease because longer lightpaths are blocked and connections tend to traverse shorter routes under higher loads. As expected, by limiting the hop count of lightpaths in PAC-HCL connections tend to traverse shorter

[2] In a sense, PAL behaves like link protection of a mesh network whereas PAC behaves like path protection.



(a) Working Paths

(b) Backup Paths

Figure 5.3: Average Hop Count of Connections

hops and hence consume less bandwidth resources; this is one of the reasons that PAC-HCL outperforms other schemes.

So far, we have neglected the effects of the network grooming capacity on the performance of the grooming schemes. Figure 5.4 shows the BBP vs. grooming capacity when the network load is 24 Erlangs. The figure shows when the number of grooming add/drop ports is smaller, PAL outperforms PAC and PAC-HCL. That is expected since a protection lightpath in a *p-lightpath* under PAL does not consume any grooming ports. Therefore, when this number is small, the poor performance of both schemes of PAC is evident (64% $\sim$ 73% blocking). However, as the number of grooming ports increases, the performance gradually improves. One notable observation is that PAC-HCL outperforms PAC, which is different than one would expect; that is, the shorter is the lightpath, the more grooming ports one needs to consume. The reason PAC-HCL shows better performance than PAC is due to the four reasons mentioned before. The bandwidth in the logical topology is more judiciously used due to the increased routing flexibility and better bandwidth

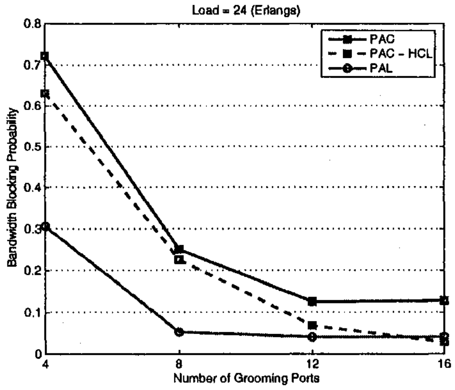


Figure 5.4: BBP vs. Grooming Capacity

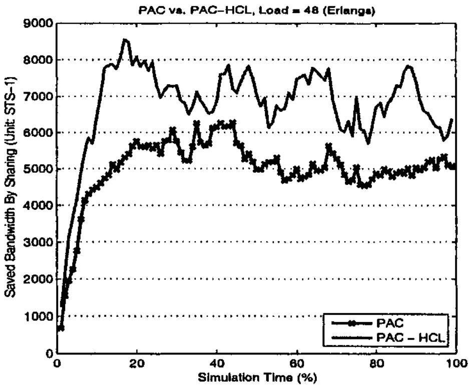


Figure 5.5: Backup Bandwidth Sharing Performance

sharing. Therefore, fewer lightpaths are setup and hence fewer grooming ports are consumed. It is important to mention here that the average nodal degree of the network studied is 3.74 and the number of wavelengths per link is 8; when the grooming capacity per node is 16, PAC-HCL outperforms the other grooming schemes.

Next we study the sharing efficiency of backup bandwidth under PAC and PAC-HCL. We do not consider PAL, since sharing is not end to end and is done between *p-lightpath*s at the wavelength level (i.e., PAL normally has better sharing). We measure the total amount of backup bandwidth reserved at a particular load (e.g., 48 Erlangs) throughout the simulation time for both schemes. The base of comparison is the dedicated protection, in which no bandwidth sharing is allowed. Here, if the set of connections protected by a particular lightpath is $c_i(i = 1,\ldots,n)$ with w_i bandwidth per demand, then the amount of backup bandwidth reserved on that lightpath to protect those demands is $\sum_{i=1}^{n}(w_i)$. If sharing is allowed, then the backup bandwidth reserved on a lightpath l is $v_l^* = max_{\forall e' \in E}\{v_l^{e'}\}$ where $v_l^{e'}$ is the bandwidth reserved on l to protect against the failure of link e' in the network; $v_l^{e'} = \sum_{i=1}^{M}(w_i), 0 \leq v_l^{e'} \leq STS-192$.

We calculate the saving that bandwidth sharing yields over the dedicated protection case under both schemes; e.g., the saving per lightpath l is $\sum_{i=1}^{n}(w_i) - v_l^*$. Clearly, as Figure 5.5 shows, the saving under PAC-HCL is more than that of PAC which means that backup bandwidth sharing is more efficient under PAC-HCL. The figure shows a maximum bandwidth of almost 3000 STS-1 that PAC-HCL can additionally save over the savings achieved by PAC. On average this additional saving is 1772 STS-1. We also find in simulation that that under PAC-HCL, more demands are admitted into the network (11.57% more than PAC) and that the bandwidth reserved to protect the connections is on average 463 STS-1 less than that of PAC. So, compared with PAC, PAC-HCL protects more demands by using less resources.

5.4.2 Reprovisioning Results

We simulate the failure of one unidirectional link and we calculate the percentage of unprotected/vulnerable connections in the network before and after reprovisioning for the two schemes.

Figure 5.6 shows the percentage of unprotected connections[3]. Clearly, under PAL the percentage of unprotected connections before reprovisioning is more than PAC and PAC-HCL (HCL=3). To understand the reason, we note here that the set of unprotected connections includes (1) connections that directly fail and (2) connections that become unprotected because they lost their protection connections. Our simulation results tell us that the higher percentage of unprotected connections in PAL is mainly coming from category

[3] The total number of connections in the network when the link fails in PAC is smaller than PAC-HCL and the latter is slightly smaller than that under PAL. Example, 2191 vs. 2340 vs. 2390 at 60 Erlangs loads.

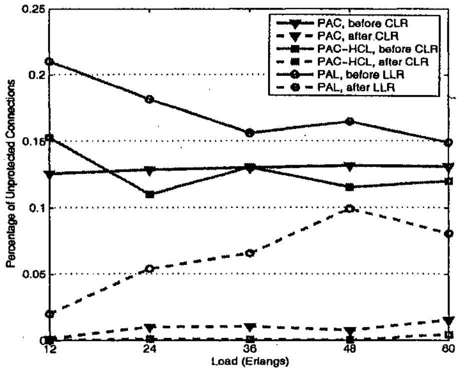


Figure 5.6: Unprotected Connections

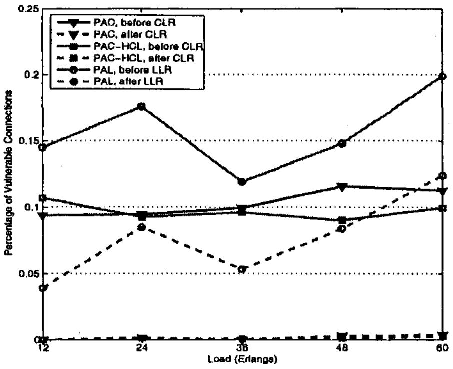


Figure 5.7: Vulnerable Connections

(2). To elaborate, note the fact that PAL has a better backup bandwidth sharing than PAC and PAC-HCL. Further, note that under PAL the percentage decreases as the load increases (e.g., from 21% to 15% as the load varies between 12 Erlangs and 60 Erlangs). But this does not necessarily mean that the number of unprotected connections at higher load is lower in the network. The reason is that at a higher load, the total number of connections admitted into the network becomes large and hence when a link fails, the fraction of unprotected demands is higher in comparison with the fraction at lower loads although the percentage is lower. PAC and PAC-HCL on the other hand shows similar results; the percentage of unprotected connections under PAC-HCL is slightly smaller, however the total number of unprotected connections between the two schemes is very close.

Now after reprovisioning, LLR yields a large number of unprotected connections by comparison with CLR. The reason that LLR does not have good performance is due to the granularity at which LLR reprovisions connections; here, only unprotected lightpaths are reprovisioned, instead of unprotected connections, by requesting resources from the physical layer. Moreover, when LLR fails to reprovision a lightpath, all connections traversing that lightpath remain exposed to a new failure. Alternatively, CLR reprovisions connections at a finer granularity than LLR. CLR exploits resources at the logical (or lightpath) layer to find sufficient protection resources. When this fails, it requests resources from the physical layer to setup new lightpaths in order to protect exposed connections. Our simulation showed that more than 80% of the unprotected connections are successfully reprovisioned using CLR at the logical layer whereas only less than 20% of connections are reprovisioned by setting new lightpaths at the physical layer.

Figure 5.6 also shows that although PAC and PAC-HCL both use CLR, PAC-HCL yields a slightly lower percentage of unprotected connections after reprovisioning. This is due to the fact that under PAC, a connection traverses a longer path (i.e., larger physical hop count) and hence consumes more network resources than PAC-HCL. Table 5.1 and Table 5.2 present the numbers of connections to be reprovisioned in physical/logical topology under PAC and PAC-HCL; columns A, B, C and D represent the number of unprotected and vulnerable connections before reprovisioning, the number of connections successfully reprovisioned in the logical topology, the number of connections successfully reprovisioned in the physical topology, and the number of unprotected connections left after reprovisioning (note that, a vulnerable connection becomes unprotected if it cannot be reprovisioned) respectively. For example, in Table 5.1, under the load of 60 Erlangs, 101 ($79 + 22$ in column C and D) connections are left to be reprovisioned at physical layer, and only 22 connections are failed to be reprovisioned and are left unprotected. Under the same load, Table 5.2 shows that more than 50% of the connections are left unprotected in PAC. Clearly, this is mainly due to higher resource availability at the physical layer under PAC-HCL.

Figure 5.7 shows the connection vulnerability before and after reprovisioning. Clearly,

Loads	A	B	C	D
12	85	85	0	0
24	170	160	9	1
36	342	309	30	3
48	374	305	65	4
60	461	360	79	22

Table 5.1: PAC-HCL

Loads	A	B	C	D
12	115	114	1	0
24	230	215	5	10
36	319	290	12	17
48	516	431	56	29
60	594	473	55	66

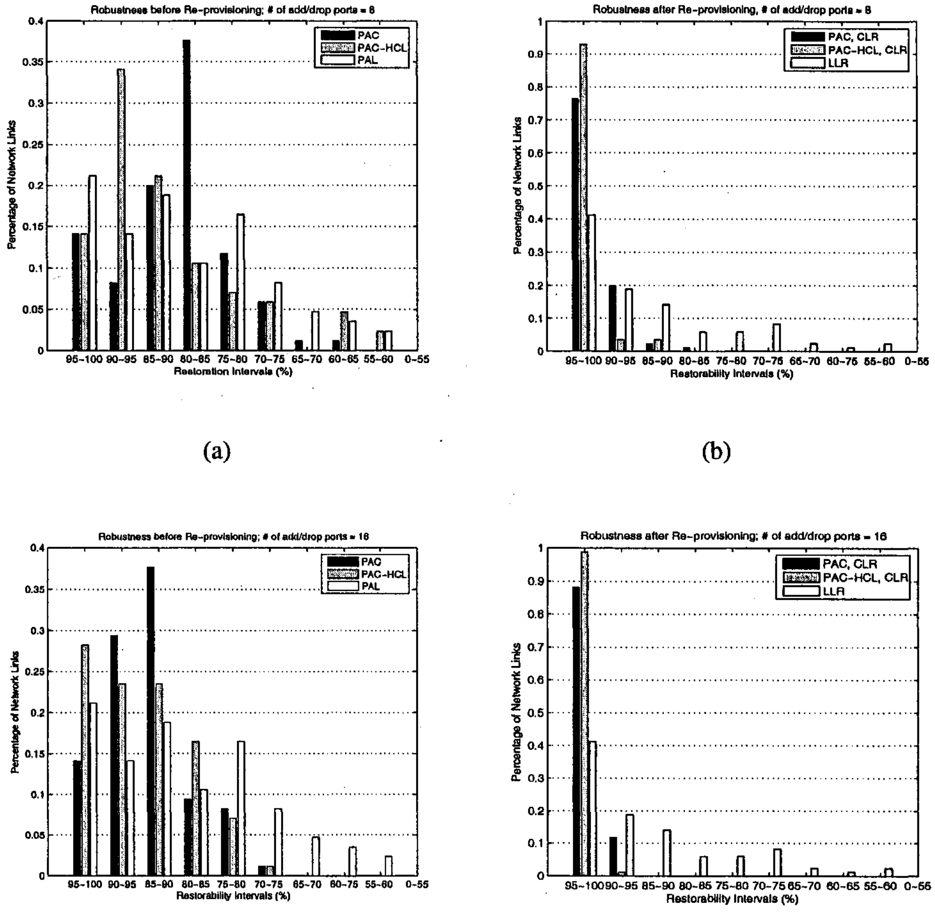
Table 5.2: PAC

the higher is the sharability of protection resources, the more would be the vulnerability of connections after the first failure. The figure shows that PAL has always higher vulnerability before and after reprovisioning, and the vulnerability increases as the load increases which is due to the fact that the sharing potential gets higher at higher loads. Similar to before, since the granularity of LLR is a lightpath, when a vulnerable lightpath fails to be reprovisioned, all connections carried by this lightpath remain vulnerable (and hence unprotected). CLR, on the other hand, due to its finer granularity substantially reduces the vulnerability of connections groomed using either PAC or PAC-HCL. We similarly notice that the majority of vulnerable connections are reprovisioned at the lightpath layer.

Network robustness is another important performance metric used to compare LLR and CLR. Figure 5.8 shows a comparison of the network robustness before and after reprovisioning at a load of 24 Erlangs. Similarly, we use 10 different intervals for the restorability ranging from $0 \sim 100\%$. First, LLR improves the robustness of PAL; before reprovisioning

$Pr(R \geq 90\%) = 0.35$ (Figure 5.8(a)) and after reprovisioning this value becomes 0.6 (Figure 5.8(b)). This is justified from Figures 5.6 and 5.7 where we showed that the percentage of unprotected connections drops from 18% to 6% (at a load of 24 Erlangs) and the percentage of vulnerable connections drops from 18% to 8% before and after reprovisioning correspondingly. Alternatively, the robustness (e.g., $Pr(R \geq 90\%)$) of PAC (PAC-HCL) improves from 22% (48%) before reprovisioning to almost 96% using CLR (Figure 5.8(a), (b)). This shows a substantial improvement of CLR over LLR; this is clearly explained in the previous discussions where after reprovisioning only a very small percentage of unprotected and vulnerable connections exist in the network. CLR performance is equal for PAC and PAC-HCL (e.g., at higher restorability) due to the small percentage of vulnerable and unprotected connections remaining in the network.

Another observation is with regards to the impact of grooming capacity on the network robustness. We study the robustness when the grooming capacity is 8 and 16 add/drop ports per node. As mentioned earlier, the grooming capacity has minor impact on PAL (see Figure 5.4) and hence on LLR. This is due to the fact that under PAL protection lightpaths do not consume any add/drop ports. However, the grooming capacity has direct effect on PAC and PAC-HCL and hence on CLR. For example, before reprovisioning when we increase the grooming capacity from 8 (Figure 5.8(a)) to 16 (Figure 5.8(c)), the robustness (e.g., $Pr(R \geq 90\%)$) changes from 22% and 48% to 43% and 51% for PAC and PAC-HCL correspondingly. After reprovisioning, the robustness of PAC and PAC-HCL changes from around 96% to almost 100% after reprovisioning (Figure 5.8(b), (d)). We find that when the grooming capacity increases, small gain is achieved by the reprovisioning algorithm (in terms of robustness). Although as shown in Figure 5.4, the BBP reduces substantially and



(a)

(b)

(c)

(d)

Figure 5.8: Average Hop Count of Connections

hence more connections are admitted into the network.

5.5 Conclusion

In this chapter we considered the problem of protecting low speed connections in optical networks against multiple near simultaneous failures. These low speed connections are groomed together either using PAL or using PAC survivable grooming policies. To improve the survivability of these connections, we proposed to use spare capacity reprovisioning after the first failure in order to allocate protection resources and protect exposed and vulnerable connections. We proposed two different reprovisioning schemes, LLR and CLR, and studied their performances. The two schemes differ in the granularity at which they reprovision spare resources and which grooming policy they each require. We have shown that CLR substantially outperforms LLR due to the increased flexibility that it enjoys. In addition, CLR reuses the available capacity at the lightpath level to protect exposed or vulnerable connections before requesting resources from the physical layer. Our results have shown that 80% of the unprotected/vulnerable connections are accommodated at the lightpath layer. LLR on the other hand only reprovisions at the physical layer although resources may be available in the existing lightpaths. We have measured the robustness of the network against dual failures and have shown that a network deploying CLR with PAC as a grooming policy achieves a very high robustness by comparison to LLR under PAL.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis investigated the survivability issues in optical networks. A series of novel models, algorithms and approaches were developed and simulated, and performance evaluations were presented by detailed simulations. This thesis made three important contributions to the body of knowledge on the design and analysis of survivable optical networks.

First, we introduced an Offset-Time based Scheduling model to achieve the goal of fast recovery against link failure(s). Experimental results showed that the new model eliminates the propagation delays and the accumulation of switching delays, and it achieves the best recovery performance upon the single link failure. Meanwhile, we analyzed its applicability under double-link failure assumption.

Second, we addressed the capacity reprovisioning mechanism and designed a new reprovisioning approach. We proved by both theoretical analysis and simulation results that this approach can protect more connections with less management overhead and resource contention. We also showed that the our proposed scheme outperforms conventional schemes under both centralized and distributed implementation. Moreover, our work considered the impact of resource sharability on the performance of reprovisioning.

Finally, we proposed two different reprovisioning frameworks for lower-speed demands in optical networks, namely, Lightpath Level Reprovisioning (LLR) and Connection Level Reprovisioning (CLR) under two grooming mechanisms, PAL and PAC. We showed that CLR substantially outperforms LLR due to the increased flexibility in reprovisioning. In addition, we found that a network deploying CLR with PAC as a grooming policy achieves a very high robustness by comparison to LLR under PAL.

6.2 Future Work

The future work can be in the following topics:

1. As stated in Chapter 5, CLR deals with a larger number of connections, therefore the management overhead may be excessive as opposed to the smaller number of lightpaths that LLR deals with. Hence, we intend to assess the overhead resulting from CLR and how this could impact the robustness of network when CLR is implemented.

2. One major requirement for optical networks is to classify customer demands based on their service availabilities. Hence, our future research will focus on designing protection mechanisms for optical networks to satisfy their availability requirements.

3. Another possible extension is to compare the path-based protection with p-cycle protection in terms of their capacity requirements and robustness to dual failures.

Bibliography

[1] B. Mukherjee, "Optical Communication Networks", *McGraw-Hill*, July 1997.

[2] T.E. Stern, K. Bala, "Multiwavelength Optical Networks A Layered Approach", *Addison Wesley*, 1999.

[3] Ming-Chwan Chow, "Understanding SONET/SDH: Standards and Applications", *Andan Publisher*, 1995.

[4] P. Michael Henderson, "Introduction to Optical Networks", *Mindspeed Technologies, Inc.*, January 29, 2001.

[5] W. Grover, "Mesh-based Survivable Networks: Optical and Strategies for Optical, MPLS, SONET and ATM networking", *Prentice Hall*, 2003.

[6] http://www.alcatel.at/products/productsummary.jhtml?relativePath=/x/opgproduct/Alcatel_1640_WM.jhtml

[7] D. J. Shin et al., "Hybrid WDM/TDM-PON for 128 subscribers using λ-selection-free transmitters", Post-Deadline Paper, *OSA Optical Fiber Communications Conference (OFC 2004)*, Los Angeles, CA, February 2004.

[8] C. Guillemot et al., “Transparent optical packet switching: the European ACTS KEOPS project approach”, *IEEE/OSA Journal of Lightwave Technology*, Vol. 16, No. 12, December 1998.

[9] P. Ashwood-Smith et al, “Generalized MPLS signaling functional description”, IETF Draft ⟨draft-ietf-mpls-generalized-signaling-06.txt⟩, April 2001.

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

[11] H. Zang, J. P. Jue, and B. Mukherjee, “A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks”, *Optical Networks Magazine*, vol. 1, no. 1, January 2000.

[12] G. N. Rouskas, “Routing and Wavelength Assignment in Optical WDM Networks”, *Wiley Encyclopedia of Telecommunications, (John Proakis, Editor), John Wiley & Sons*, June 1999.

[13] G. N. Rouskas and H. G. Perros, “A Tutorial on Optical Networks”, *Networking 2002 Tutorials*, LNCS 2497, pp. 155-193, 2002.

[14] Optical domain service interconnect. http://www.odsi-coalition.com.

[15] The optical internetworking forum. http://www.oiforum.com.

[16] Z. Zhang, J. Fu, D. Guo, and L. Zhang, “Lightpath routing for intelligent optical networks”, *IEEE Network*, 15(4): 2835, July/August 2001.

[17] C. Assi, M. Ali, R. Kurtz, and D. Guo, "Optical networking and real-time provisioning: An integrated vision for the next-generation internet", *IEEE Network*, 15(4): 3645, July/August 2001.

[18] B. Rajagopalan et al, "IP over optical networks a framework", IETF Draft ⟨draftmany-ip-optical-framework-03.txt⟩, March 2001.

[19] J. P. Lang et al, "Link management protocol (LMP)", IETF Draft ⟨draft-ietf-mplslmp-02.txt⟩, September 2001.

[20] J. Moy, "OSPF version 2. RFC 2328", April 1998.

[21] D. Awduche et al, "RSVP-TE: Extensions to RSVP for LSP tunnels", IETF Draft, ⟨draft-ietf-mpls-rsvp-lsp-tunnel-08.txt⟩, February 2001.

[22] O. Aboul-Magd et al, "Constraint-based LSP setup using LDP", IETF Draft ⟨draftietf-mpls-cr-ldp-05.txt⟩, February 2001.

[23] R. Braden et al, "Resource reservation protocol version 1. RFC 2205", September 1997.

[24] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas, "LDP specification. RFC 3036", January 2001.

[25] P. Ashwood-Smith et al, "Generalized MPLS signaling RSVP-TE extensions", IETF Draft ⟨draft-ietf-mpls-generalized-rsvp-te-05.txt⟩, October 2001.

[26] P. Ashwood-Smith et al, "Generalized MPLS signaling CR-LDP extensions", IETF Draft ⟨draft-ietf-mpls-generalized-cr-ldp-04.txt⟩, July 2001.

[27] P. Bonenfant, “Optical layer survivability: A comprehensive approach”, *Proc. OFC 1998*, Vol. 2, pp. 270271, San Jose, CA, February 1998.

[28] O. Gerstel and R. Ramaswami, “Optical layer survivability: A services perspective”, *IEEE Communications Magazine*, Vol. 38, pp. 104-113, March 2000.

[29] S. Ramamurthy and B. Mukherjee, “Survivable WDM mesh networks, part I - protection”, *Proc. IEEE INFOCOM 1999*, Vol. 2, pp. 744-751, March 1999.

[30] S. Ramamurthy and B. Mukherjee, “Survivable WDM mesh networks, part II - restoration”, *Proc. IEEE ICC 1999*, Vol. 3, pp. 2023-2030, June 1999.

[31] C. Ou, H. Zang, and B. Mukherjee, “Sub-path Protection for Scalability and Fast Recovery in Optical WDM Mesh Networks”, *Proc. OFC 2002*, pp. 495-496, March 2002,.

[32] P. H. Ho and H. T. Mouftah, “A Framework for Service-Guaranteed Shared Protection in WDM Mesh Networks”, *IEEE Communications Magazine*, pp. 97-103, February 2002.

[33] R. Ramaswami and K. N. Sivarajan, “Routing and Wavelength Assignment in All-Optical Networks”, *IEEE/ACM Transactions on Networking*, Vol. 3, No. 5, pp. 489-500, October 1995.

[34] I. Chlamtac, A. Ganz, and G. Karmi, “Lightpath Communications: An Approach to High-Bandwidth Optical WANs”, *IEEE Transactions on Communications*, Vol. 40, No. 7, pp. 1171-1182, July 1992.

[35] C. Assi, A. Shami, M. Ali, “Optical Networking and Real-Time Provisioning; An Integrated Vision For The Next-Generation Internet”, *IEEE Network*, Volume 15, Issue 4, pp. 36-45, July-Aug. 2001.

[36] C. Assi, Y. Ye, S. Dixit, and M. Ali, “Control and Management Protocols for Survivable Optical Mesh Networks, *IEEE Journal of Lightware Technology*, November 2003.

[37] **W. Huo**, C. Assi and A. Shami, “A New Framework for Rapid Restoration in Optical Mesh Networks”, *IEEE IWNDA 2004, ICPP Workshops 2004*, Montreal, Canada, August 2004.

[38] C. Assi, **W. Huo**, A. Shami, and N. Ghani, “Improving Signaling Recovery in Shared Mesh OpticalNetworks”, to appear in the *Elsevier Computer Communications Journal*, (Accepted April 2005).

[39] C. Assi, **W. Huo**, A. Shami, and N. Ghani, “Analysis of Capacity Re-Provisioning in Optical Mesh Networks”, *IEEE Communications Letters*, Vol. 9,No. 7,pp. 658-660, July 2005.

[40] C. Assi, **W. Huo**, A. Shami, and N. Ghani, “On The Benefits of Lightpath Re-Provisioning in Optical Mesh Networks”, *IEEE International Conference on Communications (ICC 2005)*, Seoul, Korea, May 2005.

[41] C. Assi, **W. Huo** and A. Shami, “Centralized vs. Distributed Re-provisioning in Optical Mesh Networks”, *The 4th International Conference on Networking (ICN 2005)*, Reunion Island, April 2005.

[42] C. Assi, **W. Huo**, A. Shami, "Impact of Resource Sharability on Dual Failure Restorability in Optical Mesh Networks", *IFIP NETWORKING 2005*, Waterloo, Canada, May 2005.

[43] **W. Huo**, L. Guang, C. Assi, A. Shami, "Survivable Traffic Grooming In Optical Networks with Multiple Failures", *18th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2005)*, Saskatoon, Canada, May 2005.

[44] **W. Huo**, C. Assi, A. Shami, "Multiple-Link Failures Survivability in Optical Networks with Traffic Grooming Capability", submitted to *IEEE International Conference on Communications (ICC 2006)*, Istanbul, Turkey, 2006.

[45] J-F. Labourdette, "Shared Mesh Restoration in Optical Networks", *Proc. OFC 2004*, February 2004.

[46] G. Li, J. Yates, D. Wang, and C. Kalmanek, "Control Plane Design for Reliable Optical Networks", *IEEE Communication Magazine*, Vol. 40, Issue. 2, pp. 90-96, February 2002.

[47] J. Yates and G. Li, "Challenges in Intelligent Transport Network Restoration", *Optical Fiber Commun. Conf.*, invited paper, March 2003.

[48] M. Goyal, G. Li, and J. Yates, "Shared Mesh Restoration: A Simulation Study", *IEEE OFC 2002*, March 2002.

[49] R. Doverspike, G. Sahin, J. L. Strand, and R. W. Tkach, "Fast Restoration in a Mesh Network of Optical Cross-connects", *Proc. OFC 1999*, February 1999.

[50] J. P. Lang et al, "Generalized MPLS Recovery Functional Specication", Internet Draft-Work, August 2002.

[51] M. Goyal, J. Yates, G. Li, and W. Feng, "Benefit of Restoration Signaling Message Aggregation", *Proc. OFC 2003*, March 2003.

[52] H. Choi, S. Subramaniam, and H. A. Choi, "On Double-Link Failure Recovery in WDM Optical Networks", *IEEE INFOCOM*, pp. 808-816, June 2002.

[53] W. He, A. Somani, "Path-based Protection for Survivable Double-link Failures in Mesh-Restorable Optical Networks", *Proc. IEEE GlobeCom 2003*, December 1-5, 2003.

[54] M. Clouqueur, W. D. Grover, "Mesh-restorable networks with complete dual failure restorability and with selectively enhanced dual-failure restorability properties", *SPIE OPTICOMM*, Boston, MA, July-Aug 2002.

[55] M. Clouqueur, W.D. Grover, "Availability Analysis of Span-Restorable Mesh Networks", *IEEE JSAC*, vol.20, no. 4, pp. 810-821, May 2002.

[56] D. Schupke, R. Prinz, "Performance of Path Protection and Rerouting for WDM Networks Subject to Dual Failures", *Proc. OFC 2003*, March 2003.

[57] S. Kim, S. Lumetta, "Evaluation of Protection Reconfiguration for Multiple Failures in WDM Mesh Networks", *Proc. OFC 2003*, March 2003.

[58] R. Ramamurthy, A. Akyamac, J-F. Labourdette, S. Chaudhuri, "Pre-Emptive Reprovisioning in Mesh Optical Networks", *Proc. OFC 2003*, March 2003.

[59] P. Charalambous, et.al., "A National Mesh Network Using Optical Cross-Connect Switches", *Proc. OFC 2003,* March 2003.

[60] J. Zhang, K. Zhu, B. Mukherjee, "A Comprehensive Study on Backup Reprovisioning to Remedy the Effect of Multiple-Link Failures in WDM Mesh Networks", *ICC 2004*, Paris, France, June 20-24, 2004.

[61] M. Clouqueur and W. D. Grover, "Computational and design studies on the unavailability of mesh restorable networks", *Proc. DRCN 2000*, pp. 181-186, Munich, Germany, April 2000.

[62] J. Doucette, M. Clouqueur and W. D. Grover, "On the availability and capacity requirements of shared backup path-protected mesh networks", *Optical Networks Magazine*, pp. 29-44, November/December 2003.

[63] D. Schupke, W. D. Grover, and M. Clouqueur, "Strategies for Enhanced Dual Failure Restorability with Static or Reconfigurable p-Cycle Networks", *ICC 2004*, Paris, France, June 20-24, 2004.

[64] O. Gerstel, R. Ramaswami, "Optical Layer Survivability- An Implementation Perspective", *IEEE JSAC,* pp. 1885-1899, vol.18, no. 10, May 2000.

[65] J. Doucette, W. D. Grover, "Maintenance-Immune Design of Span-Restorable Mesh Networks", *Proc. 18th NFOEC 2002*, Dallas, TX, September 2002.

[66] D. A. Schupke, A. Autenrieth, and T. Fischer, "Survivability of Multiple Fiber Duct Failures", *Proc. DRCN 2001.*

[67] E. Bouillet, J-F. Labourdette, R. Ramamurthy, S. Chaudhuri, “Enhanced Algorithm Cost Model to Control Tradeoffs in Provisioning Shared Mesh Restored Lightpaths”, *Proc. OFC 2002*, March 2002.

[68] L. Shen and B. Ramamurthy, “Centralized vs. Distributed Connection Management Schemes under Different Traffic Patterns in Wavelength-Convertible Optical Networks”, *Proc. IEEE ICC 2002*, NY, 2002.

[69] Y. Mei, and C. Qiao, “Efficient Distributed Control Protocols for WDM Optical Networks”, *Proc. ICCCN*, September 1997.

[70] G. Kaigala, W.D. Grover, “On the Efficacy of GMPLS Auto-Reprovisioning as a Mesh-Network Restoration Mechanism”, *Proc. IEEE Globecom 2003*, San Francisco, December 1-5, 2003.

[71] S. Sengupta, R. Ramamurthy, “Capacity Efficient Distributed Routing of Mesh-Restored Lightpaths in Optical Networks”, *Proc. GlobeCom 2001*, pp. 2129-2133, San Antonio, TX, November 2001.

[72] K. Zhu and B. Mukherjee, ”Traffic grooming in an optical WDM mesh network”, *IEEE JSAC, vol. 20*, pp. 122-133, Jan. 2002.

[73] E. Modiano and P. J. Lin, ”Traffic grooming in WDM networks”, *IEEE Communication Mag., vol. 39*, pp. 124-129, July 2001.

[74] B. Mukherjee, C. Ou, H. Zhu, K. Zhu, N. Singhal, and S. Yao, ”Traffic Grooming for Mesh Optical Networks”, *OFC 2004*, Feb. 2004.

[75] Hongyue Zhu, Hui Zang, Keyao Zhu, and Biswanath Mukherjee, "A Novel, Generic Graph Model for Traffic Grooming in Heterogeneous WDM Mesh Networks", *IEEE/ACM Trans. on Networking, vol. 11, issue 2*, pp. 285 - 299, April 2003.

[76] O. Gerstel, R. Ramaswami, and G. H. Sasaki, "Cost-effective traffic grooming in WDM rings", *IEEE/ACM Trans. Networking, vol. 8*, pp. 618-630, Oct. 2000.

[77] X. Zhang and C. Qiao, "An effective and comprehensive approach for traffic grooming and wavelength assignment in SONET/WDM rings", *IEEE/ACM Trans. Networking, vol. 8*, pp. 608-617, Oct. 2000.

[78] Canhui Ou, Keyao Zhu, Hui Zang, Laxman H. Sahasrabuddhe, and Biswanath Mukherjee, "Traffic Grooming for Survivable WDM Networks - Shared Protection", *IEEE JSAC, vol. 21, issue 9*, pp. 1367-1383, Nov. 2003.

[79] Wang Yao and Byrav Ramamurthy, "Survivable Traffic Grooming in WDM Mesh Networks under SRLG Constraints", *ICC05*, Seoul, Korea, May 2005.

[80] D. Xu, C. Qiao and Y. Xiong, "An Ultra-fast Shared Path Protection Scheme - Distributed Partial Information Management, Part II", *ICNP'02*, Paris, Nov. 2002, pp. 344-353.