EFFICIENT DESIGN AND MANAGEMENT OF RELIABLE OPTICAL NETWORKS

BY

SUN-IL KIM

B.S., State University of New York at Binghamton, 2000
M.S., University of Illinois at Urbana-Champaign, 2001

DISSEbATION

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science in the Graduate College of the University of Illinois at Urbana-Champaign, 2008

Urbana, Illinois

Doctoral Committee:

Associate Professor Steven S. Lumetta, Chair
Professor Roy H. Campbell
Professor Ravi Iyer
Professor Klara Nahrstedt
ABSTRACT

Optical networks have played a major role in allowing us to meet the bandwidth demands driven by the explosive growth of the Internet. As more and more bandwidth-hungry applications emerge, the aggregated traffic at the backbone layer will continue to grow. This increase in capacity poses challenges, especially in terms of reliability and overall management overhead. At the same time, the high bandwidth available in backbone networks lead to an increase in vulnerability to failures as huge amounts of data can be lost in a relatively short period of time. In order to meet the bandwidth demand as well as the performance requirements of future network services, cost-effective solutions that guarantee a desired level of reliability at the core must be provided. To this end, understanding the limitations of various survivability algorithms and protocols is important.

Given the cost of physically modifying the backbone, as well as the relatively slow speed at which core architectures change compared to higher layers, it is important to be able to optimize the network using soft solutions to achieve better performance. A few important aspects of network reliability and failure management are studied in this thesis. First, a study on the impact of multiple failures and techniques for addressing such failures are presented. Second, service differentiation based on reliability needs is studied. Service differentiation is critical in balancing network operation costs as well as in maximizing network utilization. Third, a cost-efficient, high-speed recovery scheme is introduced and evaluated under online non-dynamic and dynamic scenarios, as well as static provisioning.
applicable to protecting mission-critical traffic. Finally, the impact of physical
layer impairments on network reliability is presented.
To Lois
I would like to thank my advisor Steve Lumetta for his advice, guidance and support. I cannot even begin to describe everything Steve has taught me over the many years I spent as a graduate student. I am very grateful for the time and the freedom he gave me to work things out at my own pace. I can only hope to give forward and do the same for others in the future.

I would like to thank my thesis committee members for their invaluable input. I thank Prof. Roy Campbell for spending time with me to discuss issues with optical networking. I also enjoyed discussing future directions and trends in computing as well as its impact on society. I thank Prof. Ravi Iyer for challenging me to find myself in research and to grow. I am grateful to Prof. Klara Nahrstedt who always took the time to sit with me and help me with research and career directions. I am also grateful for all the research and career advice Prof. Admela Jukan has given me.

A big part of this work could not have been possible without the help of Nnam and Xiaolan. I wish to thank Chi-wei, Derek, Edwin, Gin, Hide, Hong-Seok Kim, Jeff, Kevin, Liuyang, Seung Yi and Yuan for all the interesting discussions, help and advice. I thank Dr. Tom Gambill, Andrei, Eric, Hui, Mohsen, Shadi and Yaniv from CS101 for the great three semesters I had. I also am grateful to Dr. Jeff Gaynor for being a wonderful teacher and a role model. I would also like to thank Ms. Lila Rhodes and Ms. Heidi Leerkamp.
# TABLE OF CONTENTS

| LIST OF TABLES .................................................. | ix          |
| LIST OF FIGURES ................................................ | x           |
| LIST OF ABBREVIATIONS ........................................ | xiv         |
| CHAPTER 1 Introduction .......................................... | 1           |
| 1.1 Reliability in Optical Networks ............................ | 4           |
| 1.2 Failure Models ............................................... | 4           |
| 1.3 Failure Management ........................................... | 7           |
| CHAPTER 2 Multiple Failure Management ......................... | 13          |
| 2.1 Multiple Failure Management Paradigm ....................... | 15          |
| 2.1.1 1+N Pre-allocation ......................................... | 17          |
| 2.1.2 Static Reconfiguration ..................................... | 17          |
| 2.1.3 Dynamic Reconfiguration .................................... | 18          |
| 2.2 Supporting Graceful Degradation ............................ | 19          |
| 2.2.1 Minimally Overlapping Path Selection .................... | 21          |
| 2.2.2 Robustness Metric ......................................... | 23          |
| 2.2.3 Failure Model and Failure Management ..................... | 24          |
| 2.2.4 Filtering: Robust Path Selection .......................... | 25          |
| 2.2.5 Dynamic Protection Reconfiguration ....................... | 30          |
| 2.2.6 Optimal Protection Reconfiguration ....................... | 33          |
| 2.3 Evaluation ................................................... | 38          |
| 2.3.1 Network and Traffic Model .................................. | 38          |
| 2.3.2 Results .................................................... | 40          |
| 2.4 Summary ........................................................ | 45          |
| CHAPTER 3 Reliability-Differentiated Management ............... | 47          |
| 3.1 Protection-based Service Differentiation .................... | 49          |
| 3.1.1 QoS Under WDM Networks .................................... | 49          |
| 3.1.2 Survivability ............................................... | 50          |
| 3.2 Protection Differentiated QoS ............................... | 53          |
| 3.2.1 Protection Classes ......................................... | 53          |
| 3.2.2 Classification Scheme ...................................... | 55          |
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Multiple Failure Management Schemes Using Protection Algorithms</td>
<td>15</td>
</tr>
<tr>
<td>2.2</td>
<td>Definitions for Optimal Reconfiguration</td>
<td>35</td>
</tr>
<tr>
<td>2.3</td>
<td>Evaluation results for LATAX network. Maximum two-link failure vulnerability for LATAX is $</td>
<td>E</td>
</tr>
<tr>
<td>2.4</td>
<td>Evaluation results for eight sample networks (Measurements after failure and recovery of a single link and before the second link failure). ARPANET’s Vul. and AFS can be reduced to 0 using the filtering algorithm because it is a 3-connected network</td>
<td>42</td>
</tr>
<tr>
<td>2.5</td>
<td>Reconfiguration cost results for eight sample networks. * symbol denotes ILP problems too large to run on our computers</td>
<td>44</td>
</tr>
<tr>
<td>3.1</td>
<td>Protection based QoS classes</td>
<td>51</td>
</tr>
<tr>
<td>3.2</td>
<td>Protection based QoS classes</td>
<td>52</td>
</tr>
<tr>
<td>3.3</td>
<td>Total capacity cost for different classification schemes normalized to class S1</td>
<td>58</td>
</tr>
<tr>
<td>3.4</td>
<td>Avg # of failed lightpaths / Avg. # of lightpaths affected by a link failure (avg. link load)</td>
<td>61</td>
</tr>
<tr>
<td>4.1</td>
<td>Simulation Results</td>
<td>111</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1.1 Basic taxonomy of survivability schemes. .......................... 7
1.2 Illustration of different protection approaches. ...................... 10

2.1 Example of two different failure classes shown on the FunLim graph. 19
2.2 Minimally overlapping path selection. After a link failure ($(A, D)$) and reconfiguration, the connection is left vulnerable only to failure of link $(A, B)$. .......................... 19
2.3 Simple algorithm for computing *minimally overlapping paths*. This algorithm is optimal for finding a backup path that has the minimal cost among maximally robust paths. Complexity for this algorithm using is $O(|E| + |N| \log |N|)$ (same as Dijkstra’s algorithm). .......................... 20
2.4 Reconfiguration scheme overview. (a*)Figure 2.6. (b*)Figure 2.10. (c*)Figure 2.3. .......................... 21
2.5 Non-robust path pairs. Example of non-critical link failure affecting two-link survivability due to poor initial routing choice. . . . 26
2.6 Algorithm for filtering out poor solutions. Complexity for computing critical links is $O(|N|^2|E|^2d_{M_2})$, $d_{M_2}$ is second largest node degree, using Ford-Fulkerson algorithm for function $f$. Complexity for computing non-robust path pairs is $O(|N|^2|CP_M|\frac{|E|}{2}d_{M_2})$, $|CP_M|$ is the maximum number of disjoint path pairs between a node pair given some hop limit. It is important to note that both algorithms are used for initialization and run only once for each topology. .......................... 28
2.7 Ratio of non-robust and total number of primary-backup path pairs for SÚBNJ. Arbitrary length backup paths are utilized. . . . 29
2.8 Ratio of non-robust and total number of primary-backup path pairs using shortest primary paths. .......................... 30
2.9 Ratio of non-robust and total number of primary-backup path pairs using primary paths that are up to 1 hop longer than the shortest path(s). .......................... 31
2.10 Outline of a simple DPR scheme with minimally overlapping path selection. Complexity of the steps outlined here is $O(C(|E| + |N| \log |N|))$ (the maximum capacity of a link in terms of the number of wavelengths in the network, $C$, times the complexity of running MOP.) ........................................ 32

2.11 Sample Networks. ................................................................................. 39

3.1 The National network. ................................................................. 58
3.2 Example Network ......................................................................... 59
3.3 On-line capacity provisioning results on the National network. . . 60
3.4 Capacity for Class A and Class B traffic with varying percentage of Class A traffic for National. Arrows point to data at 1:2 ratio corresponding to data shown in Figure 3.3. ................................. 61

4.1 A) SPP setup B) Example of a Streams setup ......................... 66
4.2 Streams: PXC configuration. .................................................... 68
4.3 A working example of the Streams protection scheme. .......... 70
4.4 Networks used for evaluation. .................................................... 78
4.5 Approximate data loss and recovery times. The $\delta$ terms denote the propagation delay on the backup path, which may vary slightly between the different protection schemes, but has no effect on data loss figures. ...................................................... 81
4.6 Total capacity normalized to DPP with shortest primary paths for link failure protection. ................................................................. 85
4.7 Average backup path lengths in hops with shortest primaries. PL-expansion numbers are shown inside the parentheses. .. 86
4.8 Operating points for the different algorithms: Normalized capacity and pl-expansion. The lowest points represent shortest (+0 hop) backup with data points corresponding to additional hops from the shortest. .................................................. 87
4.9 Total capacity normalized to DPP with shortest primary paths for link and node failure protection. ................................. 89
4.10 Average backup path lengths in hops with shortest primaries. PL-expansion numbers are shown inside the parentheses. .... 89
4.11 Operating points for the different algorithms: Normalized capacity and pl-expansion. The lowest points represent shortest (+0 hop) backup with data points corresponding to additional hops from the shortest. ...................................... 90
4.12 Total capacity from online provisioning for link failures. 98% confidence intervals are shown for 0-hop and N-hop solutions. DPP cost is always the same since the shortest path pairs yield the lowest cost. ......................................................... 91
4.13 Total capacity from online provisioning for link and node failures. 98% confidence intervals are shown for 0-hop and N-hop solutions. DPP cost is always the same since the shortest path pairs yield the lowest cost. ........................................ 91
4.14 Single link failure protection. Blocking probability vs. traffic load (Erlangs). .............................................................. 94
4.15 Single link failure protection. Primary and backup path lengths (normalized to the average shortest path length) vs. traffic load (Erlangs). .............................................................. 95
4.16 Single link and node protection. Blocking probability vs. traffic load (Erlangs). .............................................................. 97
4.17 Single link and node protection. Primary and backup path lengths (normalized to the average shortest path length) vs. traffic load (Erlangs). .............................................................. 98
4.18 (a) Simple path-stream. (b) Simple cycle-stream. (c) Complex cycle-stream. (d) Complex path-stream. ......................... 101
4.19 (a) Two compatible primaries. (b) Incompatible primaries. (c) Incompatible primaries. ............................................. 105
4.20 (a) both backup paths of two primaries crossing (b) compatible assignment (c) incompatible assignment .................. 108
4.21 Networks ................................................................. 109
4.22 Q-stream efficiency ..................................................... 109

5.1 AON Node architecture. ................................................... 117
5.2 Strictly non-blocking 8x8 switch architecture built using 2x2 couplers. The table shows the number of copies of NxN Banyan required to build a strictly non-blocking switch. Three 8x8 Banyan crossconnect design is used to meet the requirements of a strictly non-blocking switch. By design, the combiners and splitters before and after the core Banyans are disjoint among lightpaths. Therefore, crosstalk (1st order) occurs only in the core. ................................................................. 118
5.3 Demux/mux crosstalk. ..................................................... 119
5.4 Crosstalk is introduced in the switch between primary path P2 and the backup path B1 when B1 is turned on after the link failure. ................................................................. 123
5.5 Call admission process. (* RWA is performed according to the different algorithms described in this section.) .......................... 126
5.6 Crosstalk counting for potential backup path selection for SPP. All possible failure cases must be enumerated. One case shown: link (1,2). ................................................................. 129
5.7 15N-Mesh and NJLATA. .................................................. 137
5.8 15N-Mesh DPP. ............................................................. 139
5.9 15N-Mesh SPP. ............................................................. 140
5.10 NJLATA DPP. ........................................... 143
5.11 NJLATA SPP. ........................................... 144
5.12 a) FR for 15N-Mesh DPP(top) & SPP(bottom) b) FR for
      NJLATA DPP(top) & SPP(bottom) ..................... 146
### LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AON</td>
<td>All-Optical Network</td>
</tr>
<tr>
<td>DLP</td>
<td>Dedicated Link Protection</td>
</tr>
<tr>
<td>DPP</td>
<td>Dedicated Path Protection</td>
</tr>
<tr>
<td>DWDM</td>
<td>Dense Wavelength Division Multiplexing</td>
</tr>
<tr>
<td>ILP</td>
<td>Integer Linear Programming</td>
</tr>
<tr>
<td>OEO</td>
<td>Optical-Electronic-Optical</td>
</tr>
<tr>
<td>OPR</td>
<td>Optimal Protection Reconfiguration</td>
</tr>
<tr>
<td>OXC</td>
<td>Optical Crossconnect</td>
</tr>
<tr>
<td>PON</td>
<td>Passive Optical Network</td>
</tr>
<tr>
<td>PXT</td>
<td>Photonic Crossconnect</td>
</tr>
<tr>
<td>QoT</td>
<td>Quality of Transmission</td>
</tr>
<tr>
<td>RWA</td>
<td>Routing and Wavelength Assignment</td>
</tr>
<tr>
<td>SLP</td>
<td>Shared Link Protection</td>
</tr>
<tr>
<td>SPF</td>
<td>Shortest Path First</td>
</tr>
<tr>
<td>SPP</td>
<td>Shared Path Protection</td>
</tr>
<tr>
<td>SRLG</td>
<td>Shared Risk Link Group</td>
</tr>
<tr>
<td>TON</td>
<td>Transparent Optical Network</td>
</tr>
<tr>
<td>WDM</td>
<td>Wavelength Division Multiplexing</td>
</tr>
</tbody>
</table>
Rapid advances and widespread use of high bandwidth networks, as well as mobile/ubiquitous services, are constantly reshaping the way we access information and leading to development of many new applications. As evidenced by the explosive growth of data traffic in recent years, as more and more network systems are deployed to meet the needs of emerging applications, the amount of data aggregated at the core infrastructure will continue to increase at an alarming rate. In order to meet the bandwidth demand as well as the performance requirements of future network services, the core network architectures must also evolve. Management complexity will also grow as a result, necessitating scalable management techniques that are more efficient. Thus far, optical networks have allowed us to keep up with the bandwidth demands (currently, systems have been demonstrated the capability to support over one hundred wavelength-channels at as high as 80Gbps per channel over a single optical fiber). However, the current core architectures exhibit many shortcomings in terms of performance and flexibility. Many important issues in cross-layer design, management and integration of next generation optical networks need to be carefully studied in order to better support future communication needs. To this end, examining the performance tradeoffs in cost, reliability and management overhead is critical.

One of the most critical issues in designing and managing optical backbone networks is reliability. While allowing us to support an unprecedented amount of data traffic, the high data rates exacerbates the impact of failures. A huge
amount of data and revenue can be lost in a short amount of time in the event of a failure. For example, in a system supporting 32 40Gbps channels in each direction on an optical fiber, a down time of mere 1 second of down time results in a loss of over 2.5Tb of data. Therefore, it is important to understand the impact of different types of failures and provide efficient solutions to address them. In this work, several important aspects of managing optical networks are addressed with a focus on reliability.

First, we provide an insight into how backbone networks need to be managed as they become more heterogeneous, providing services for applications (or higher layer networks) with diverse reliability needs. Given that minimizing capacity cost (measured in the number of wavelength-channels provisioned in the network) is critical in efficient management of a network, various recovery techniques that vary in cost and recovery-speed can be utilized to serve the varying needs of different classes of traffic.

Second, we study the impact of multiple failures on networks designed with single failure recovery mechanisms. We introduce new metrics that can be used to quantify the impact of multiple failures and measure how well traditional approaches perform under multiple failures. We then propose a combination of management techniques technique, based on a small amount of preplanning and quick post-failure reconfiguration, that allow a network to gracefully degrade under such failures.

Third, we design a cost-efficient network protection algorithm and protocol, called Streams, with the goal of rapid recovery. Using Streams, we showed that cost-efficient end-to-end protection can be achieved without sacrificing recovery speed. Prior to our approach, utilization of end-to-end protection schemes always resulted in penalty in recovery speed or cost overhead. We evaluate the performance of the Streams protection scheme under online non-dynamic and dynamic
routing scenarios and compare the results with the fastest recovery scheme as well as the most cost-efficient recovery scheme. We also show that the Streams protection scheme can be used to design protected core networks for high-priority connections using static, offline provisioning. We propose a novel optimization technique using integer linear programming based on a novel heuristic that allows us to effectively reduce the problem space. Finally, we quantify the impact of physical layer impairments on reliability of all-optical networks. We evaluate quality of transmission (QoT) aware routing algorithms with end-to-end protection, and quantify the overhead in guaranteeing 100% survivability under transmission impairments. We also quantify the performance of best-effort approaches and show that its application depends heavily on the topology of the network. Using simulation with detailed models of physical layer components, we also draw a comparison between using a hardware-based approach (additional hardware is used to overcome some of the problem with signal crosstalk) and a “soft” solution based on smarter routing techniques.

The rest of this thesis is organized as follows: First, a discussion of various failure models in optical networks as well as a survey of previous work are provided. The topic of multiple failure management is presented in detail in Chapter 2. Chapter 3 covers reliability-differentiated protection along with an optimization technique that can be used in wavelength routed multi-class networks. In Chapter 4, we present a high-speed recovery algorithm called Streams. In Chapter 5, we present a study on the impact of physical layer impairments (caused by imperfect characteristics of network components such as amplifiers and switch fabrics) on network reliability in the context of optically transparent networks. Finally, a summary and discussion of future directions is provided in Chapter 6.
1.1 Reliability in Optical Networks

Various failure models are used to study the impact of network failures, as well as in designing efficient management techniques to address the failures. In this chapter, I first provide a brief overview of the common failure models that are used throughout this work. I then present a short survey of the previous work in optical network survivability.

1.2 Failure Models

General failure modes in optical networks consist of channel failures, link failures, and failures of optical crossconnects (OXC). Channel failures are the most common, and are often caused by the failure of a line card or cards at a port of a switching node. Network connections utilizing a failed channel become disconnected, and the higher layer networks or applications start to lose data. Higher layer networks or protocols, such as the well-known Internet Protocol (IP), may have fault-resilience built into them. However, failures at the backbone level can leave networks physically disconnected, rendering higher-layer protocols helpless.

Link failures are also common, and lead to failure of all channels on all fibers in the link. Some of the most common causes for link failures include amplifier failures, fiber cuts caused by wayward backhoes and switch port failures. Most of the work in the area of network survivability focus on link failures as the same protocols and algorithms can be used to handle channel failures. The underlying assumption here is that the failures are detected/initiated by the end nodes on a per-channel basis. There are fiber-based failure detection and recovery initiation protocols (discussed in the next section), but most such schemes do not allow efficient use of network resources in multi-channel WDM networks.
Network nodes, which consist of optical crossconnects (OXC) for networks that utilize optical-electronic-optical conversion (OEO) or photonic crossconnects (PXT) for networks that utilize all-optical components, can also fail. Complete node failures, caused by hardware failures, power outages, malicious attacks, etc., are much less common, but can cause a much more severe problem as all network links and channels that originate, pass through, or terminate at the failed node go down at the same time. This type of failure leaves the network in a much more vulnerable state given that it is equivalent to several links failing at the same time. Given the relatively small size of the network topologies at the WDM layer, removing several links leaves the network heavily disconnected. Recovery from complete node failures for traffic originating or terminating at the failed node requires the use of hardware redundancy where the entire node is duplicated, but networks are seldom built to this level of reliability. However, one interesting thing to note is that it is easier to replace failed switches or parts of the failed node compared to a fiber that has been cut. Depending on the type of the problem, link failures can take longer to physically repair.

Due to the small probability of multiple failures, and to minimize the network resources that need to be reserved to handle failures (overhead), survivability schemes are usually designed to handle single failures. However, the ability to degrade gracefully in the event of multiple network failures is important in meeting the goals of the design and management of reliable networks. The significance of multiple failure models is that they can be used to understand the impact of failures that can not be captured using single failure models [1, 2], thereby allowing us to quantify the network’s ability to achieve graceful degradation as well as allowing us to design better management techniques.

Multiple failures can be separated into two categories, differentiated by the temporal relationship between failures. Simultaneous failures refer to cases where
multiple components fail close enough in time to disrupt normal recovery (in networks designed to handle only single failures). It is highly unlikely to have two independent failures fall within the general recovery time goal for rapid recovery, which is in the order of 100ms or less. We discuss the recovery timing in the next section. However, a network may experience another type of simultaneous failures, called shared risk link group failure (SRLG), such as a cut through a conduit shared by several topologically diverse links. SRLGs require special attention as all links in the SRLG fail in the event that SRLGs take place. Generally, SRLGs are hard to handle without the information on their location, and pose complicated management issues. Often, SRLG detection is difficult in itself [3]. On the other hand, if SRLG information is given, different provisioning techniques can be used to avoid SRLG failures from disrupting connections for long periods of time. The provisioning method is effective (again, given SRLG information) in preventing such failures from ever occurring, but is much more constrained in route selection and optimization and poses a big challenge [4, 5, 6, 7, 8]. SRLG issues are not covered in this thesis as the techniques presented in the literature are orthogonal to the various methods of survivability techniques described in this work, and therefore can be used together.

Sequential failures are failures of multiple components separated by enough time for the recovery algorithm to complete recovery of a single failure, but before this failure can be physically replaced or repaired. For many rapid recovery schemes, recovery times are on the order of milliseconds (typically 20 to 100 milliseconds [9]) in the optical layer, whereas physical repair of failed components may take several hours to days/weeks. It is easy to see that the networks are exposed to subsequent failures while waiting for physical repairs to happen. Given the relative time scales for the recovery protocol and waiting for physical repair to take place, sequential failures are much more likely to occur compared to si-
multaneous failures. Due to the fact that it is much more likely to occur than simultaneous failures, sequential failures have been previously used to study the impact of multiple failures in [1, 2].

### 1.3 Failure Management

To protect the networks from failures discussed in the previous section, various recovery techniques have been proposed in the past. The key tradeoffs amongst these techniques are recovery-speed, the cost (defined as the amount of extra network capacity required to implement a particular recovery scheme), management overhead and compatibility (in terms of special hardware use). In this section, a brief survey of the previous work in the literature and examples that explain an intuitive classification of recovery schemes are presented.

A general taxonomy of survivability schemes is shown in Figure 1.1. In general, survivability schemes are classified into two main categories—Protection and Restoration—in [10]. Restoration locates free wavelength-channels ($\lambda$-channels) for backup after a failure occurs, and therefore requires minimal amount of extra capacity to be reserved for backup. Protection preplans backup routes that are used in the event of a failure. Protection and restoration offer a tradeoff between
the speed of recovery and efficiency in terms of the use of spare capacity [11, 12]. Restoration schemes are more efficient in terms of capacity requirements, and offer better multiple failure survivability because they dynamically find backup paths after a failure. However, protection can also be implemented in a capacity efficient manner [13, 14, 15, 16, 17, 18, 19] and offer much faster recovery with the absence of the excess signaling delay needed for dynamic route discovery [20, 21, 22]. Restoration based recovery takes about 2 seconds (time spent searching for new routes as well as connection re-establishment), whereas protection schemes can achieve complete recovery in the order of tens of milliseconds [23]. For example, Synchronous Optical Networks (SONET) have a 50ms protection switching capability and falls under protection. As many of the networks are built on top of SONET and have become dependent on reliable operation with rapid-recovery, protection schemes are assumed for this work. It is also important to note that rapid recovery schemes will becomes increasingly important as we become more and more dependent on reliable operation of underlying communication networks.

Protection algorithms can be further classified into local (link) protection and path (end-to-end) protection. Path protection requires selection of disjoint primary and backup path pairs. In 1+1 (one-plus-one) protection, traffic is actively sent on both paths, and the receiving node simply switches to the backup path in the event of a failure [23]. This type of protection offers fast recovery with little or no data loss because no signaling is required between the source and the destination nodes, but is inefficient in terms of capacity requirements. It is also expensive to actively drive two live paths for every connection. With 1:1 protection, dedicated backup channels are also reserved for each primary channel, but the backup path does not carry live traffic until there is a failure. The backup paths are then capable of carrying additional unprotected (and preemptible) traffic. This reuse makes 1:1 even more efficient than 1+1 protection in terms of capacity utiliza-
tion; the tradeoff is the additional delay before the traffic is switched onto the backup path, which increases the recovery time compared to the 1+1 approach. These end-to-end protection schemes are refereed to as dedicated path protection (DPP). However, 1:1 is assumed in this work due to operation costs. Even though recovery times are slower, the amount of data loss between 1+1 and 1:1 is the same. I provide a discussion on the recovery timing later in this section.

In shared end-to-end protection schemes, called shared path protection (SPP), backup channels are chosen in advance, but not preconfigured. Instead, the end nodes of a lightpath signal the intermediate nodes to establish the backup route after a failure occurs. Capacity reserved for backup can be shared among different connections that do not share nodes or links in their primary paths (hence cannot fail at the same under the single failure assumption), or can be used to carry low priority (unprotected) traffic, which is preempted in the event of a failure. The need to signal and configure intermediate nodes renders SPP slow compared to DPP, but SPP requires the least amount of reserved capacity.

Link protection schemes react more quickly to failures than do path protection schemes since failures are detected by the end-nodes of the failed link rather than the end-nodes of the entire path. Recovery is also initiated by the same nodes. The difference between dedicated protection and shared protection for link protection schemes is the same as the difference between DPP and SPP described above. Dedicated link protection (DLP) is very similar to DPP except for the failure detection and recovery initiation procedure. Many shared link protection (SLP) schemes are similar to shared path protection in the sense that backup capacity is reserved but not preconfigured. The nodes at the end of the failed link signal and configure the intermediate nodes after the failure. The drawback is that link protection is significantly less efficient than path protection in terms of capacity usage as more capacity needs to be reserved for backup [16, 18, 24].
Some protection schemes offer a balance between capacity and recovery speed of the link and path approaches by dividing up the path protection approach into a number of domains (or segments) [25, 26].

Figure 1.2 illustrates the four general protection schemes. All four examples show two routed connections from node $a$ to node $d$. We denote the paths $p_1$ ([a-b-c-d]) and $p_2$ ([a-g-h-d]). In DPP, $p_1$ and $p_2$ share the same recovery route [a-e-f-d], but separate wavelengths are reserved. In SPP, $p_1$ and $p_2$ share both the recovery route and a backup wavelength channel. DLP and SLP protects each individual link along the primary path using separate recovery routes. Link [a,b] is protected by path [a-e-b] (Similarly-[b,c]-[b-e-f-c], [c,d]-[c-f-d], [a,g]-[a-e-g], [g,h]-[g-e-f-h], [h,d]-[h-f-d]). DLP pre-allocates separate wavelength channels for use by recovery routes, whereas SLP allows sharing of wavelengths channels to better optimize capacity usage.
The importance of these four protection schemes is that they capture the characteristics of most protection algorithms found in the literature. For example, DLP based on selecting shortest recovery paths possible for each link provides the fastest recovery among all protection algorithms, including mesh protection [10, 15, 19], generalized loopback [27], p-cycles [28], ring-based schemes such as cycle double covers [29], and other protection methods that attempt to offer advantages of both PP and LP [25, 26]. On the other hand, SPP is considered the most capacity-efficient protection algorithm [9].

Some of the specific examples of rapid recovery schemes include the use of logical ring embeddings in a mesh network. Protection schemes that use rings, such as cycle double covers [29], provide rapid recovery through the use of Automatic Protection Switching (APS), which automatically switches traffic over to protection fibers in the event of a failure. Ring-based solutions, however, pose difficult optimization problems of finding ring covers, and do not guarantee 100% connectivity between all pairs of nodes with protection against node failures [30] without complex extensions to enable backup paths to hop between rings. They are also inefficient in terms of protection capacity, requiring between 100% and 300% additional capacity [28]. The p-cycles work [28] solves the capacity problem for rings while providing fast recovery, using preconfigured cycles to protect against failures of links in both existing and newly designed networks. Flow p-cycles [31] extends the concept of p-cycles to path segments and provides protection for both link and node failures. The flow p-cycles algorithm is more capacity-efficient compared to the p-cycles as claimed in [31]. Both p-cycles and flow p-cycles leverage hop-to-hop OEO conversion. P-cycle configurations for networks with partial wavelength conversion has also been studied [32]. Generalized loopback uses a flooding-based approach to provide link protection, giving a more flexible and efficient implementation than many ring algorithms while providing rapid, APS-like recovery [27].
As with many link protection schemes, generalized loopback decouples primary routing from protection allocation, and performs well when link loads are fairly uniform. With non-uniform link loading, however, generalized loopback requires substantially more protection capacity.
CHAPTER 2

Multiple Failure Management

The number of critical applications that depend on reliable operation of backbone networks are increasing, necessitating methods that allow the networks to maintain a high level of robustness. As both the size and complexity of these networks continue to increase, the ability to gracefully degrade in the event of a failure becomes important. To this end, addressing multiple failure survivability is critical. As discussed in the previous chapter, there are many survivability techniques that offer tradeoffs between recovery speed, protection capacity, and management overhead and complexity. Most of these techniques, however, focus on single failure models. In order to maximize the robustness of a network, and to allow graceful degradation, multiple failure models must be considered. Recently, the interest in understanding such failures and studying efficient methods of addressing them has been increasing [1, 2, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46].

In this chapter, techniques for maximizing the robustness of a network under sequential failures are presented. First, based on some basic failure models, a simple classification scheme for multiple failure management techniques is introduced. Then, an algorithm that allows for a selection of maximally robust working and protection path pairs is presented. An algorithm that separates out poor routing choices is also presented and a discussion on the importance of assigning robust paths during the initial provisioning stage is provided. These two techniques, combined with protection reconfiguration, allows us to meet the goal of designing a scheme that efficiently protects the network against sequential failures.
A greedy heuristic is utilized to perform *dynamic protection reconfiguration* (DPR) to achieve fast reconfiguration of recovery routes and re-allocate backup wavelengths after a failure (online) to allow better recoverability from subsequent failures. An ILP solution is formulated to understand the optimal (minimum) cost. Reconfiguration in general is an efficient way to address sequential failures as the capacity needed to reallocate the connections that are affected by the initial failure is relatively small. Previous studies have looked at handling reconfiguration in a way that allows the network to recover from subsequent failures only if there is a fully disjoint path available for affected connections. Some have also focused specifically on double-link failure models (specifically, preplanning for double-link failures). Because many networks found in practice are not very well connected (most are two-link connected but not three-link connected), these approaches leave many connections unprotected between recovery from the first failure and the time when physical repair is made. Using the techniques presented in this chapter, a network can be made more robust; in practice, the robustness is limited by the topology and should not be hindered by poor routing choices or failure management techniques.

Management techniques that are used together to achieve the maximum robustness for a given topology are presented. First, we introduce a filtering algorithm that eliminates poor primary and backup path pair choices that prevent optimal reconfiguration. We then propose the use of what we term *minimally overlapping paths* (MOP) for protection reconfiguration after failure. Using the filtering algorithm and MOP selection along with multiple failure management schemes (we focus on reconfiguration, but applicable to pre-allocation schemes as well) guarantees maximum robustness under sequential multiple failures. Understanding failure dynamics with a detailed study using probabilistic models for link failures is also important and can be found in [47, 48]. The scope of this work is
different and is in designing efficient techniques to manage/handle such failures when they occur. We address the need for simple and effective mechanisms.

The remainder of this chapter is organized as follows. The next section provides an outline of multiple-failure management schemes. In Section 2.2, we present the techniques that can be used to maximize network robustness, and provide a simple, greedy heuristic as well as an integer linear programming (ILP) solution for DPR. We then discuss the experimental setup and present the results from simulation runs in Section 2.2. A short summary is provided in Section 2.4.

### 2.1 Multiple Failure Management Paradigm

To study multiple failures, we consider sequential two-link failures. A two-link failure consists of two independent link failures in a network graph. As defined in the previous chapter, the second failure occurs long enough after the first to allow normal recovery to complete but before any physical repair can be accomplished. The two-link failure model effectively captures the characteristics of sequential failures, and aids in understanding some of the important issues in multiple failure survivability. It also allows us to understand the effect of general multiple failures on a network rather than to simply study a specific, two link failure scenarios. For example, considering recovery between two sequential link failures can be used to

<table>
<thead>
<tr>
<th>RWA</th>
<th>1+N Pre-allocation</th>
<th>Static Reconfiguration</th>
<th>Dynamic Reconfiguration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline</td>
<td>Online w/ online reoptimization</td>
<td>Online</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Complexity</th>
<th>High (ILP)</th>
<th>Moderate (heuristic) to High (ILP)</th>
<th>Low (fast heuristic)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Capacity Cost</th>
<th>High (1+N paths)</th>
<th>Moderate (2 paths+$\Delta$)</th>
<th>Low (~2 paths)</th>
</tr>
</thead>
</table>

| Management Overhead | Low | Low to moderate | Moderate |

Table 2.1: Multiple Failure Management Schemes Using Protection Algorithms.
think about what happens between the second and the third failures under three-link failures. Thus, it is important to keep in mind that the ideas and findings presented in this work are not limited to two-link failures, and can be applied to any number of sequential failures in general. However, the probability of network partitioning increases steeply with more than two failures.

We classify multiple failure management schemes based on protection algorithms into three categories—1+N pre-allocation, static reconfiguration, and dynamic reconfiguration. Table 1 summarizes the tradeoffs between the different classes of schemes. 1+N pre-allocation (offline multiple failure protection planning) techniques pre-allocate backup wavelengths using precomputed recovery routes based on a static traffic demand. The remaining two classes of algorithms fall under reconfiguration. Reconfiguration refers to multiple failure protection management schemes that rearrange/reallocate protection resources after a failure occurs, so that additional failures can be handled. Static Reconfiguration performs optimization and allocation of these extra recovery paths online at the time of provisioning. Protection resources can be reoptimized each time a new connection arrives for better resource utilization, but it can introduce a considerable amount of management complexity as a difficult optimization problem may need to be solved frequently. To the best of our knowledge, online multiple failure protection planning has not been studied. Dynamic protection reconfiguration adapts dynamically to failures by computing and allocation resources for new recovery routes (in preparation for a second failure) after a failure occurs and is recovered by the original protection scheme (only for the connections that were affected by a failure). We next present a more detailed discussion off these schemes.
2.1.1 1+N Pre-allocation

Pre-allocation involves setting up 1+N diverse paths (where N is the number of failures that the network must handle) and assigning wavelengths for all connections. One drawback of preplanning is that it can only be used to protect against a fixed number of failures. For example, to survive a failure of up to three links, three protection paths must be provisioned. This scheme best supports static traffic, and thus allows for offline capacity optimization of routing and wavelength assignment (RWA) [35, 34, 33].

Integer Linear Programming (ILP) approaches are most commonly used to optimize for capacity, but often are slow for large networks. Full optimization may not be possible for many practical networks as the complexity of optimization problems grows exponentially with the size of the network and the demand as well as a change in N. This problem is not a shortcoming of the 1+N Pre-allocation technique, but a challenge in complex optimization problems in general. The optimization process, however, does not pose a serious problem in terms of management overhead, as it is performed only once for the network. Heuristics, such as genetic algorithms and simulated annealing, can be used, but designing efficient heuristics may also be difficult. Protection capacity requirement is high for pre-allocation. Existing techniques have focused on optimizing capacity for double link failures and require significantly more capacity compared to the cost for routing working paths (over 200% capacity) [33, 35, 38, 41].

2.1.2 Static Reconfiguration

In static reconfiguration, two diverse paths are computed and allocated for each connection (one for primary and the other for backup), and some buffer wavelengths are reserved based on a computation of all possible two-failure scenarios.
Capacity cost therefore takes into account wavelengths required for allocating the two paths plus some extra (buffer wavelengths, \( \Delta \)) as shown in Table 1. Optimization for RWA can be done using ILP or other heuristics depending on the network requirements. In the event of a failure, after the initially affected connections are recovered, the network is reconfigured according to the precomputed second failure scenario using the buffer wavelengths. Management can then choose to reiterate the buffer computation and reservation process in order to support recovery from additional failures. During this reoptimization process, either ILP or some heuristic can be used again. Capacity cost is lower compared to pre-allocation, but still may be expensive depending on the efficiency of the optimization technique used. Static reconfiguration schemes are useful especially for networks that are built on leased lines with service agreements for a specific number of \( \lambda \)-channels (wavelength channels). For these networks, it may not be possible to quickly ask for and obtain additional \( \lambda \)-channels or fibers from the main network, necessitating the pre-allocation of the buffer wavelengths. To the best of our knowledge, no static reconfiguration allocation algorithm has been designed yet.

### 2.1.3 Dynamic Reconfiguration

Since most protection schemes are designed to handle one failure at a time, they can be extended to handle sequential failures using dynamic reconfiguration. In the event of a failure, dynamic reconfiguration identifies and protects the failed connections and the connections that are left vulnerable to additional failures. Reconfiguration information is then dynamically computed. Because the network dynamically adapts to a specific failure, dynamic reconfiguration can handle an arbitrary number of sequential failures (as long as the topology permits), and requires little additional capacity. It is important to understand that capacity
Figure 2.1: Example of two different failure classes shown on the FunLim graph.

Figure 2.2: Minimally overlapping path selection. After a link failure \((A, D)\) and reconfiguration, the connection is left vulnerable only to failure of link \((A, B)\).

utilization of dynamic reconfiguration can be at least as good as other schemes because dynamic reconfiguration only uses resources that are required to address a specific failure to maximize the robustness of a network.

### 2.2 Supporting Graceful Degradation

In order to achieve maximum robustness for a given network, we consider MOPs (minimally overlapping paths), where the number of overlapped links between
The goal is to find a protection path $P_b$ that has a \textit{minimal overlap} with the working path $P_w$. Use any weighted shortest path algorithm with link cost $c_l$ defined as follows for each link $l \in E$:

$$c_l = \begin{cases} |N| & \text{if } l \in P_w \\ 1 & \text{otherwise} \end{cases}$$

If the $\lambda$-channel sharing is considered during the path search step, the cost is defined as follows:

$$c_l = \begin{cases} |N| & \text{if } l \in P_w \\ 0 & \text{if the } \lambda\text{-channel can be shared} \\ 1 & \text{otherwise} \end{cases}$$

Figure 2.3: Simple algorithm for computing \textit{minimally overlapping paths}. This algorithm is optimal for finding a backup path that has the minimal cost among maximally robust paths. Complexity for this algorithm using is $O(|E|+|N|\log|N|)$ (same as Dijkstra’s algorithm).

the live path and the backup path is minimized. Given a primary path $p$, a minimally overlapping (backup) path $b$ is a path that shares the fewest number of links with $p$ and has the same end points as $p$. Assuming a \textit{completely disjoint path} (CDP) (or non-overlapping paths) $c$ exists for $p$, $c$ is then an MOP for $p$ as well. However, in considering two-link failures, because many mesh networks are not three-connected, CDP triples do not exist for many end node pairs. For these connections, reconfiguration will be unsuccessful, leaving them completely vulnerable to additional failures. However, any multiple failure combination that does not leave the two end nodes physically disconnected can be partially protected by using MOPs that are not CDPs. Note the two different types of failures shown in Figure 2.1. Figure 2.1(1) shows examples of disconnection failures, where parts of the network become physically disconnected. No recovery technique can address this type of failure and thus the failure is termed fundamental failure. Figure 2.1(2) shows two examples of \textit{non-interruptable failures}. Non-interruptable failures occur when the network has enough capacity to provide two disjoint paths, but due to the fact that the path carrying live traffic cannot be reconfigured (without causing service disruption), post-failure network reconfig-
Routing − Upon arrival of a connection request, cheapest primary and backup path pair is assigned using the information saved during the network design step.

Network Design − Robust primary and backup path pairs (for all end node combinations) are precomputed using our filtering algorithm (a*). This step is performed only once for a given network topology and the results are stored in a database.

Online Provisioning

Network Setup Stage

Operational Stage

Figure 2.4: Reconfiguration scheme overview. (a*)Figure 2.6. (b*)Figure 2.10. (c*)Figure 2.3.

2.2.1 Minimally Overlapping Path Selection

Utilizing our filtering technique (Section 2.2.4) along with MOPs in reconfiguration allows the network to achieve maximum robustness under the given failure model and topology for protection schemes. In other words, other than fundamental failures, the networks are only subject to non-interruptible failures. Non-interruptible failures are extremely rare. The algorithm for computing MOPs is simple and is outlined in Figure 2.3. This algorithm works with a chosen primary path to find a shortest MOP backup for that primary. Figure 2.4 illustrates how our techniques fit together in managing and operating a network. During the
network setup and design stage, our filtering technique is used. Again, this step is
done only once unless there is topological change. Once the network goes online,
our overall scheme only affects network operation in that primary and backup
path pair selection is done from the database obtained during the network de-
sign stage using our filtering algorithm. Our filtering algorithm only saves robust
primary-backup pairs and, therefore, the number of choices is reduced, speeding
up route selection during online computation (over not using our method). When
there is a failure, our DPR technique is used along with MOP selection to provide
maximum robustness for the network.

Note that the filtering algorithm (Section 2.2.4) is run just once at the initial
provisioning stage of the network and dynamic reconfiguration (Figure 2.10) is
performed when there is a failure. MOP selection algorithm replaces any algo-
rithm that can be used to find shortest disjoint paths upon reconfiguration, and is
run for each connection that requires a new backup path during reconfiguration.
The complexity of the filtering technique is the least important as it is performed
just once for a network topology. Second, the complexity of the reconfiguration
algorithm depends on the network load (especially in terms of the number of con-
nections that are affected by a failure) and is bounded by the maximum capacity
on a link (because this figure is the maximum possible number of connections that
can go down when a link fails). Finally, the MOP selection algorithm does not
add complexity to commonly used technique that simply reroute by attempting
to find a completely disjoint path using a weighted shortest path algorithm of the
network manager’s choice. With MOP, we simply change the weights.
2.2.2 Robustness Metric

A two-link failure robustness measure called *vulnerability*, which measures the number of links, upon their failure, will cause some connection(s) to fail without recovery, was introduced in [2]. The vulnerability metric effectively captures the network robustness under two-link failures, but was originally intended to measure networks with routing and protection at link-level granularity (fiber switching). This metric is extended to path based protection by incorporating the notion of failure of individual connections. Vulnerability can be computed by the following formula. For two distinct links \( i \) and \( j \), \( f_i(j) \) is 1 if any connection cannot be recovered after failure of \( j \) following failure and recovery of \( i \).

\[
Vulnerability \equiv \frac{1}{|E|} \sum_{i \in E} \sum_{j \in E, j \neq i} f_i(j) \tag{2.1}
\]

Vulnerability basically is a network-wide metric measuring what portion of the network the overall robustness depends on. It gives us an idea of how much of the network (in terms of averaged number of links) will affect overall robustness if an additional failure is to occur before the first failure is physically repaired. Therefore, a low vulnerability figure suggests that the network can mostly remain fully operational after a second link failure, whereas high vulnerability suggests that a second link failure is likely to bring down at least some part of the network.

We also use a metric, called *failure susceptibility* (FS), that quantifies the network’s ability to handle failures in more detail [45]. Susceptibility measures the number of connections that are unprotected (left without protection) from subsequent failures, after failure and recovery of the first failure. Initially, when all network links are available (no failures), \( FS \) (failure susceptibilities) for all links are zero. After the failure and recovery of some link(s), the robustness of networks will diminish due to both algorithmic limitations as well as topological
constraints, and FS effectively captures this limitation. Average FS (AFS) then is the average FS over all links. Note that FS/AFS depends on the traffic load (since it counts the number of connections that cannot be protected), and therefore it is only meaningful to compare different scenarios/algorithms with the same traffic load. For two-link failure scenarios, AFS of a network can be computed according to the following formula shown below. $FS_k(l) =$ the number of connections unrecoverable upon failure of $l$ after recovery of the first link failure ($k$).

$$AFS \equiv \frac{1}{|E|(|E|-1)} \sum_{i \in E} \sum_{j,j \neq i} FS_i(j)$$  \hspace{1cm} (2.2)

Again, the AFS metric quantifies how many connections are susceptible to future failures. It is averaged over all possible two-link failure scenarios. Note that recovery ratio (or failure ratio, which is $1 - \text{recovery ratio}$) can be easily obtained by dividing AFS by the total number of connections affected by the failure. Both metrics depend upon the actual routed traffic and capture network robustness in different ways. Therefore, knowledge of the traffic load is needed. However, the metrics are not defined in terms of nor are limited to use with a specific traffic matrix.

### 2.2.3 Failure Model and Failure Management

It is important to understand that the two metrics given in the previous section can be readily extended to include different failure rates for network links, given some specific failure rates for each link. This extension can be done by simply adding a coefficient to the metrics to multiply vulnerability or susceptibility for each link pair combination with specific failure rates.

The focus of this work, however, is not to measure and explore the space of different failure rates and the impact of different failure dynamics. The goal of this
work, as mentioned previously, is to understand and address how multiple failures can be better managed if and when they occur. For reliable network operation, the different types of failures need to be managed somehow regardless of how and when they happen. The decision to implement certain types of survivability techniques is up to the network management, given that they have all the options and consider the tradeoffs. Therefore, simple/general methods for efficiently handling different types of failures are needed. As previously mentioned it is generally understood that sequential failures are much more likely to occur given reasonable failure rates. Simultaneous failures on the other hand are difficult to handle and mostly requires preplanning or rerouting (unless, of course, when it is a fundamental failure) [4, 5, 6, 7, 8]. Handling this type of failure can be complementary to our work and is outside the scope of this work. Sequential failures, however, can be handled more efficiently (compared to rerouting or preplanning) and addressing this issue as effectively as possible is the goal of this work.

As mentioned in the previous chapter, our work provides algorithms and mechanisms for protection and reconfiguration of the network. If the network management is able to eliminate some low risk link failure combinations, they can simply choose to allocate less resources for the low risk failures. Again, having such information is undoubtedly helpful, but does not change how general protection mechanisms are designed. Instead, the two different areas of focus with respect to multiple failures are complementary in the overall goal of efficiently addressing network robustness.

2.2.4 Filtering: Robust Path Selection

As discussed in the previous section, it is important to note that initial routing choices can also affect reconfiguration, and in turn, the robustness of a network.
Figure 2.5: Non-robust path pairs. Example of non-critical link failure affecting two-link survivability due to poor initial routing choice.

In many cases, a poor initial routing choice can prevent a connection from being protected after a single link failure even if a CDP triple exists (for a three-connected node pair) between the end nodes of a connection in the graph. For example, in two-link failure scenarios, an assigned working and protection path pair can prevent reconfiguration algorithms from providing maximum protection from additional failures. Because it is very unattractive to interrupt live traffic, reconfiguration algorithms only reallocate the backup paths (If primary paths are reconfigured, live connections are disrupted even when it is not affected by a failure). It is possible to have the best potential backup path blocked after the initial failure, and we say that the working and protection path pairs are non-robust if they prevent the network from reconfiguring to maximum robustness. To help understand the difference between robust and non-robust path pairs, we provide the following example.
In Figure 2.5, we show a possible choice of primary and backup path pair between nodes 2 and 9. Primary path [9-10-4-2] is one of the shortest paths between nodes 2 and 9. Assume that backup path [9-8-7-0-2] is chosen to facilitate capacity sharing with some other path. This primary-backup pair leaves the network susceptible to additional failures upon failure of link (4,10). None of the choices allow reconfiguration to protect the connection fully (resulting in sub-optimal robustness for this topology). The new protection path in choice 1 (shown in dashed line with shallow arrow head) shares link (9,8) with the live path. The same conflict can be seen in choices 2 and 3 on links (10-7) and (10-8) respectively. Note that all failure combinations not including critical links should be recoverable. The link (4-10) is not a critical link for node pair 2 and 9, yet this particular choice of primary-backup path pair prevents optimal reconfiguration for the network.

We introduce a simple algorithm (shown in Figure 2.6) that filters out non-robust path pairs, so that the working and backup path pairs can be chosen wisely to allow maximum survivability from future failures. As discussed earlier, this algorithm can be run just once during the network design stage (initialization) where the results can be saved for use during online provisioning.

Figures 2.7-2.9 show the results from applying the above algorithm to different networks (assuming all links are available, therefore representing initial stage of provisioning without failures). The network topologies are shown in Figure 2.11. The charts show the total number of disjoint path pairs for each node pair, as well as the number of non-robust pairs and the ratio of the two. They are ordered in increasing percentage of non-robust pairs. For example the 21 bars in Figure 2.7 represent 21 node pairs in the network. The gray bars represent the total number of disjoint path pairs for each node pair and the black bars show how many of those path pairs are non-robust as determined by our algorithm. They are
E - set of all links in the graph
N - set of all nodes in the graph
F - set of failed/unavailable links. Could be ∅ (no failures)
    or any size set representing the current network state.
R - set of node pairs R = N × N
P_i - set of all path for some pair i,
    where i ∈ R and ∀p ∈ P, p ⊆ E
f(i, S) - function returns maximum integer flow for pair i
    on subgraph S, where i ∈ R, S ⊆ E
CL_i - set of critical links for node pair i
CP_i - all disjoint path pairs, w/ some hop limit, for i, i ∈ R
    CP_i = {(k^1, k^2), k^1 ⊆ E, k^2 ⊆ E, k^1 ∩ k^2 = ∅}
NonRobust_i - all non-robust path pairs for i, NonRobust_i ⊆ CP_i
m_i - maximum integer flow for pair i

Compute critical links for all node pairs

1: for all i ∈ R do
2:   m_i ← f(i, E \ F)
3: for all l ∈ E do
4:   n^l_i ← f(i, E \ (\{l\} ∪ F))
5: if n^l_i < m_i then
6:   CL_i ← CL_i ∪ \{l\}

Find non-robust path pairs for all node pairs

1: for all i ∈ R do
2:   for all (k^1, k^2) ∈ CP_i do
3:     for all l ∈ k_1 ∪ k_2 do
4:       if l ∈ k^1 then
5:         temp ← f(i, E \ (k^2 ∪ \{l\} ∪ F))
6:       else
7:         temp ← f(i, E \ (k^1 ∪ \{l\} ∪ F))
8:     if l ∈ CL_i then
9:       if temp < m_i - 2 then
10:          NonRobust_i ← NonRobust_i ∪ \{k\}
11:     else
12:       if temp < m_i - 1 then
13:          NonRobust_i ← NonRobust_i ∪ \{k\}

Figure 2.6: Algorithm for filtering out poor solutions. Complexity for computing
critical links is O(|N|^2|E|^2d_{M_2}), d_{M_2} is second largest node degree, using Ford-
Fulkerson algorithm for function f. Complexity for computing non-robust path
pairs is O(|N|^2|CP_M|\sum_{E|2} d_{M_2}), |CP_M| is the maximum number of disjoint path
pairs between a node pair given some hop limit. It is important to note that both
algorithms are used for initialization and run only once for each topology.
arranged in an increasing order of fraction of non-robust path pairs (shown by the gray line). Note that all path pairs for some node pairs are non-robust in the NATIONAL, LATA and ARPANET networks when shortest primary paths are used. Therefore, we must utilize paths that are at least one hop longer than the shortest for the connections between those two node pairs in order to guarantee maximum robustness. Figure 2.9 reflects this small change for the three networks.

Note that the figures are in decreasing order of average node degree of the networks. It is visually clear from the graphs that, in general, as the average node degree decreases, the likelihood of making poor routing decisions becomes greater. Therefore, the benefit of the filtering algorithm varies with the average node degree. The cost of running the algorithm is extremely small (less than a few seconds, depending on the topology, on a desktop with Athlon 1.5Ghz processor and 512MB of RAM), especially as the results can be saved from running the algorithm just once for a given network topology. The complexity of the algorithm is given in Figure 2.6. At the same time, running the filtering algorithm can only help the robustness of a network.
2.2.5 Dynamic Protection Reconfiguration

In this section, we describe our simple dynamic protection reconfiguration algorithm that can be used in conjunction with most existing protection schemes to provide optimal multiple sequential failure protection. The goal is to maximize the network’s ability to handle multiple failures (and therefore minimize service disruptions) while taking into account capacity costs and management complexity.

As a more capacity-efficient alternative to multiple failure protection planning (1+N Pre-allocation), DPR is designed to dynamically adapt to failures. Originally, we explored the idea of quick, dynamic reconfiguration and hinted at the different tradeoffs in using such technique in [36]. An overview of similar
approaches (reconfiguration or reprovisioning) has been studied in [42]. DPR is different from a few related works published since the introduction of the multiple failure evaluation in [1]. First, [35, 33, 34, 39] solves two-link survivability in the context of link protection, but the dynamic protection reconfiguration technique is orthogonal to the choice of protection schemes. In this work, however, we focus on end-to-end recovery (path protection). Furthermore, our goal is not to provide a detailed optimization technique for a particular protection scheme per se, but rather to study and understand the tradeoffs involved in attempts to maximize network survivability under multiple sequential failures. Second, many have assumed that the network is three-link connected when studying two-link failures since three-link connected topologies allow complete recovery under any two link failures. DPR does not limit the number of failures up to which the network can survive. For instance, a network can fully handle $n$ sequential failures, if the network is at least $n + 1$-connected. Employing the dynamic protection reconfiguration techniques using MOP selection and filtering out non-robust path pairs.
The goal is to quickly reconfigure connections that are affected by a failure. Note that recovery paths redirect traffic between the source and destination nodes in path protection.

1. After a failure, identify the three sets of connections rendered vulnerable to additional failures. These sets correspond to the three classes of algorithmic failures presented in [2, 36].
   - \( C_{ph} \) = set of connections with broken primary paths (already recovered via original protection path)
   - \( C_{bp} \) = set of connections with broken recovery paths
   - \( C_{bsp} \) = set of connections for which the assigned protection wavelengths were used for recovery of traffic in \( C_{ph} \).

2. Release the wavelengths used by the original primary paths in \( C_{ph} \) and the original backup paths in \( C_{bp} \).

3. For each connection in \( C_{ph} \), \( C_{bp} \), and \( C_{bsp} \), find new paths that minimally overlap with the current working path. In order to maximize robustness, use the filtering algorithm with updated maximum flows between each node pair. (Note: These connections are reconfigured in maximum contention first ordering similar to [42]. Connections that have a common failure mode in their working path are said to be in contention, and therefore maximum contention first ordering selects a connection(s) with the most number of contentions first.)

4. Select a path for each of the affected connections from the choices found in Step 3 that minimizes resource usage. [For quick reconfiguration, we walk through the list of affected connections in random order, and greedily attempt to maximize resource sharing in assigning wavelengths.]

Figure 2.10: Outline of a simple DPR scheme with minimally overlapping path selection. Complexity of the steps outlined here is \( O(C(|E| + |N|\log|N|)) \) (the maximum capacity of a link in terms of the number of wavelengths in the network, \( C \), times the complexity of running MOP.)

as we have proposed, allows a network to reach the upper bound in survivability (limited by the topology). Lastly, others consider a static offline routing and perform offline optimization. We consider an online routing model where offline optimization is less useful.

The goal is to reconfigure the affected connections as quickly as possible while efficiently utilizing capacity. Ideally, reconfiguration should complete as fast as possible (order of seconds). This duration consists of the computation time and the network setup time. Figure 2.10 shows an outline of the steps involved in dynamic protection reconfiguration. New protection paths are found by using the same path selection technique used for originally provisioning lightpaths in the
network and starts immediately after a failure is detected. We assume that a centralized manager keeps track of information on all connections and that the affected connections are easily identified once the failure is recovered and located. Keeping a small data structure that tracks all the primary paths and the recovery paths provisioned on each link is useful in quickly identifying connections that are affected by the failure. Once the failure is detected, a simple lookup can be performed using the data structure.

2.2.6 Optimal Protection Reconfiguration

Optimal protection reconfiguration (OPR) provides the optimal solution (in terms of the number of extra $\lambda$-channels used) for paths re-selection and wavelength assignment for reconfiguring all connections that are affected by a failure. An OPR solution for a particular failure is obtained by solving an NP-hard mixed integer linear programming problem (ILP). A large amount of memory and computational power is required to obtain OPR solutions in practice (depending on the size of the network and the traffic load) and may be unattractive in many cases as reconfiguration must be completed as quickly as possible. However, OPR solutions provide lower bounds for the cost of reconfiguration algorithms and also may be used in reoptimization once the network is protected from additional failures (via a faster reconfiguration algorithm).

In this section, we illustrate an ILP formulation for OPR based on the failure information. OPR aims at selecting new protection paths for connections that are left vulnerable to additional failures. First we define some notation, shown in Table 2.2. Let $E$ be the set of edges and $W$ be the ordered set of available $\lambda$-channels on each edge. Note that the $\lambda$-channels which are currently occupied by up-to-date primaries (working paths), including the backups for primaries affected
by the failure, are not included in $W$ since they are not available and carry active traffic. The $\lambda$-channels occupied by backup paths of connections unaffected by the failure are included. $R$ is the set of all connections. $A \subseteq R$ is the set of all connections that are affected by the failure ($C_{ph} \cup C_{bp} \cup C_{bsp}$ from Fig. 10). $U \subseteq R$ is the set of all connections not affected by the failure. Note $R = U \cup A$.

Conflicting connection pairs are denoted as a pair of connections who share a common link in their primary paths (and are affected simultaneously when the common link fails). These pairs cannot share a $\lambda$-channel for their backup paths. Let $C \subseteq R \times R$ be the set of connection pair tuples that conflict with each other. Let $b_r$ be the unaffected backup path for each unaffected connection $r \in U$. Let $B = \bigcup_{r \in U} b_r$ be set of all unaffected backup paths. Note that the $\lambda$-channels used by $b_r$’s are available for use by other backup paths as long as there is no conflict in the primaries as described above. Let $f_r$ be either primary or backup path for $r \in A$ that are no longer used (a link is broken by the failure or some part of the backup path is now being used by another connection). This set obviously excludes the broken link and the $\lambda$-channels now being used to carry live path as part of backup paths for failed connections. Let $F = \bigcup_{r \in A} f_r$ be the set of all freed paths. There is no additional cost for using the $\lambda$-channels in $f_r$’s. Maximal sharing of $\lambda$-channels with paths in $B$ and $F$ is desired in order to reduce the overall capacity cost. $CB_r$ is an ordered set of candidate backup paths (either CDP or MOP) for affected pairs $r \in A$.

To simplify the formulation, we introduce some functions. $tr(e, w, P)$ returns a subset of paths $S \subseteq P$ for some path set $P$. Each path $p \in S$ traverses edge $e \in E$ and wavelength $w \in W$. $l(p)$ returns the set of edges in the path $p$. $\lambda(p, e)$ returns the wavelength assignment on edge $e$. $ch(p)$ is a tuple $(l(p), \lambda(p, e))$ of the particular edge and wavelength that are assigned to path $p$. 

34
Table 2.2: Definitions for Optimal Reconfiguration.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$</td>
<td>set of edges.</td>
</tr>
<tr>
<td>$W$</td>
<td>ordered set of $\lambda$-channels not carrying live traffic.</td>
</tr>
<tr>
<td>$R$</td>
<td>set of all connections.</td>
</tr>
<tr>
<td>$A$</td>
<td>set of connections affected by the failure.</td>
</tr>
<tr>
<td>$U$</td>
<td>set of unaffected connections.</td>
</tr>
<tr>
<td>$C$</td>
<td>set of conflicting connection pair tuples.</td>
</tr>
<tr>
<td>$b_r$</td>
<td>backup path for the unaffected connection $r \in U$.</td>
</tr>
<tr>
<td>$B$</td>
<td>set of $b_r$’s.</td>
</tr>
<tr>
<td>$f_r$</td>
<td>freed path of the affected connection $r \in A$.</td>
</tr>
<tr>
<td>$F$</td>
<td>set of $f_r$’s.</td>
</tr>
<tr>
<td>$CB_r$</td>
<td>ordered set of candidate backup path (CDP or MOP) for affected connections $r \in A$.</td>
</tr>
<tr>
<td>$tr(e, w, P)$</td>
<td>function returns a subset of path of $P$ traversing edge $e$ and wavelength $w$.</td>
</tr>
<tr>
<td>$l(p)$</td>
<td>function returns the set of edges traversing by path $p$.</td>
</tr>
<tr>
<td>$\lambda(p,e)$</td>
<td>function returns the wavelength assigned to path $p$ on edge $e$.</td>
</tr>
<tr>
<td>$ch(p)$</td>
<td>function returns the tuple ($l(p)$, $\lambda(p,e)$).</td>
</tr>
<tr>
<td>$X_{ep}$</td>
<td>binary variable taking 1 when the $w$th wavelength on edge $e$ is assigned to the $p$th candidate path of affected connection $r$.</td>
</tr>
<tr>
<td>$Y_{ew}$</td>
<td>binary variable is 1 when the $w$th wavelength on edge $e$ is assigned to some path. 0 otherwise.</td>
</tr>
<tr>
<td>$Z_{rp}$</td>
<td>binary variable is 1 when $p$th candidate path for affected pair $r$ is selected. 0 otherwise.</td>
</tr>
</tbody>
</table>

Finally we define some ILP variables. $X_{ep}$ is a binary variable set to one when the $w$th wavelength on edge $e$ is traversed by the $p$th candidate path of a affected pair $r$, where $p \in CB_r, r \in A$. $X_{ep}$ zero when the resource is not occupied by path $p$. $Y_{ew}$ is a binary variable indicating the occupancy of $\lambda$-channels. $Y_{ew}$ is one when the $w$th wavelength on edge $e$ is taken by some path and zero for free. $Z_{rp}$ is a binary variable indicating the path selection. $Z_{rp}$ is one when the $p$th candidate path for affected pair $r$ is chosen and zero when not chosen. Table 2.2 summarizes these notations.
The ILP formulation is as follows. The objective function minimizes the total number of extra λ-channels that must be allocated for reconfiguration.

\[
\text{Minimize} \quad \sum_{e \in E, w \in W, (e, w) \notin \text{ch}(p)} Y^{ew}, \text{ where } p \in B \cup F
\] (2.3)

The following two constraints enforce that two primaries that have the same link(s) along the path are assigned to separate λ-channels for their backups to prevent sharing (because they can fail at the same time and need separate resources to recover at the same time). These connections are said to be conflicting. There are two cases for conflicting connections. The first case is that one pair is unaffected and the other is affected. Then the affected pair cannot share the λ-channels with the known backup paths of the unaffected. For all \((r_1, r_2) \in \mathbb{C}, \) if \(r_1 \in U, r_2 \in A, \) then for all \(e \in E, w \in W, (e, w) = \text{ch}(b_{r_1}), \)

\[
\sum_{p \in \text{tr}(e, w, CB_{r_2})} X^{ew}_{r_1p} = 0
\] (2.4)

The second case is that both connections are affected. Thus on the λ-channels traversed by some candidate paths of both pairs should only be assigned to at most one of them. If \(r_1 \in A, r_2 \in A, \) then for all \(e \in E, w \in W, \)

\[
\sum_{p \in \text{tr}(e, w, CB_{r_1})} X^{ew}_{r_1p} + \sum_{p \in \text{tr}(e, w, CB_{r_2})} X^{ew}_{r_2p} \leq 1
\] (2.5)

Each λ-channel is counted once when multiple paths share the λ-channel.

For all \(e \in E, w \in W, \)
\[ Y_{ew}^{e_\lambda} \leq \sum_{r \in A} \sum_{p \in \text{tr}(e, w, CB_r)} X_{rp}^{e_\lambda} \]  
(2.6)

\[ Y_{ew}^{e_\lambda} \sum_{r \in A} |CB_r| \geq \sum_{r \in A} \sum_{p \in \text{tr}(e, w, CB_r)} X_{rp}^{e_\lambda} \]  
(2.7)

Each path is only assigned to one wavelength on each link it traverses.

For all \( r \in A, p \in CB_r, e \in l(p), \)

\[ Z_{rp} = \sum_{w \in W} X_{rw}^{e_\lambda} \]  
(2.8)

Only one path amongst all computed paths for each affected connections are selected as the new backup path. For all \( r \in A, \)

\[ \sum_{p \in CB_r} Z_{rp} = 1 \]  
(2.9)

The above ILP formulation can be easily converted into a pre-computed recon-figuration version, where all link failures are considered by expanding the variables to one more dimension and the sum of all extra \( \lambda \)-channels used for all links is minimized. Specifically, \( X_{rp}^{e_\lambda} \) becomes \( X_{rlp}^{e_\lambda} \), where \( l \in E \), represents the recon-figuration variable under the failure of link \( l \). Similarly, we have \( Y^{eul} \) and \( Z_{rlp}^{l} \). For each link failure, above constraints are formulated according to the failure information, such as affected connections, unaffected pairs and candidate paths. The objective function is then changed to sum up all \( Y^{eul} \), i.e.,

\[ \text{Minimize } \sum_{l \in E} \sum_{e \in E, w \in W, (e, w) \notin ch(p)} Y^{eul}, \text{ where } p \in B^l \cup F^l \]  
(2.10)

Note that the associated failure information \( B^l \) and \( F^l \) varies with the failed link \( l \). Based on the pre-selected original primary and backup path pairs for each
connection, the expanded ILP formulation will yield the optimal secondary backup solution for all failure cases and maximize the sharing of $\lambda$-channels amongst non-conflicting pairs. However, the new problem set becomes $|E|^2$ times larger than the original one. Therefore, an ILP solution for pre-computed reconfiguration may not be attractive in practice.

2.3 Evaluation

This section provides the results from our work. We study DPR based on both MOP as well as CDP using seven representative networks shown in Figure 2.11 and the simple network we used to illustrate fundamental limits shown in Figure 2.1 (FunLim). We next provide the evaluation details before discussing the results.

2.3.1 Network and Traffic Model

To some extent, failure impact depends on the network traffic conditions. Therefore, the study of different protection algorithms requires a fair and consistent basis for comparison. We assume uniform traffic demands, which can effectively aid in capturing the tradeoffs of the algorithms presented in this work. We also consider an on-line provisioning model with uniformly distributed full-mesh traffic demands (consistent with other related works found in the literature [19]), and we assume that the network is optically opaque and capable of full wavelength conversion at each hop. On-line provisioning means that we have no knowledge of future demands, and cannot reroute existing connections on the network to optimize provisioning upon receipt of a new request. Each request is assumed to be a bidirectional connection with a uniformly distributed demand of one connection between each source and destination. $(N \times (N-1))/2$ bidirectional requests are routed in random order to simulate an on-line provisioning process. Although, in
practice, the demands may not be uniformly distributed among different requests, we believe that studying uniformly distributed traffic demands is effective in that it sufficiently shows the tradeoffs in using our proposed technique for the purpose of studying multiple failure survivability. We assume that each $\lambda$-channel has a cost of 1 in terms of calculating capacity. The total cost of capacity is therefore the sum of the overall of working paths and the total number of the reserved protection wavelengths. Although a uniformly distributed traffic demand is assumed for evaluation in this work, the failure classification scheme, and the algorithms we presented assume nothing about the traffic model.

We use shared path (end-to-end) protection and greedily select a link-disjoint path pair, between the source and destination nodes, based on capacity cost for allocating each path pair. In other words, paths are chosen to minimize its allocation cost assuming no knowledge of future connection requests. Wavelength assignment for a backup route is determined by evaluating all possible available wavelengths to maximize sharing (minimize cost). With the on-line routing model, it is also assumed that no previously routed lightpaths can be disrupted to per-
form rerouting optimizations. The path selection process is similar to the joint selection algorithm presented in [15], where all choices of primary and backup paths within the path length requirements are considered. Note that a pool-based reservation [17] for backup wavelengths can also be used along with our approach without loss of efficacy in terms of robustness. However, we assigned specific wavelengths because, upon different link failures, computing sharing information for reconfiguration is simpler and faster if each backup path is assigned to specific wavelengths (because compatibility between different backup paths change when some backup paths become active and some become obsolete due to a failed link and assigning specific wavelengths can significantly reduce the number of connections that need to be considered for compatibility checks).

As discussed in Section 2.2.5, because we hope to restore lightpaths after two links are cut, we must choose the primary/backup path pair that allows another link-disjoint path. Again, this constraint virtually has no effect on routing. Connections with either degree two source or destination nodes, or both, do not have three CDPs. For DPR-CDP, they result in capacity failures. For DPR-MOP, we use the MOP selection technique presented in the previous section.

### 2.3.2 Results

Tables 2.3 shows the results for the LATAX network. They are arranged in a way that makes it easier to visualize the effect of the filtering algorithm as well as the MOP selection. First, AFS and vulnerability are shown with and without the filtering when the network is not reconfigured after the first failure. When the network is not reconfigured, over 20 connections are lost when a second failure occurs, and the network is vulnerable to 45.9 links on average. Note that the maximum two-link failure vulnerability for a given network is $|E| - 1$. 

Second, filtering benefits both CDP and MOP cases in terms of both AFS and vulnerability. As mentioned in the previous section, this benefit can vary depending on the network topology as well as the traffic load. Using the filtering algorithm always yields a solution that is at least as good as provisioning without filtering when reconfiguration is used.

<table>
<thead>
<tr>
<th></th>
<th>AFS</th>
<th>20.39</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Vulnerability</td>
<td>45.9</td>
</tr>
<tr>
<td>No Reconfig.</td>
<td>Not Filtered</td>
<td>Filtered</td>
</tr>
<tr>
<td>CDP</td>
<td>AFS</td>
<td>1.02</td>
</tr>
<tr>
<td></td>
<td>Vulnerability</td>
<td>13.15</td>
</tr>
<tr>
<td>MOP</td>
<td>AFS</td>
<td>0.19</td>
</tr>
<tr>
<td></td>
<td>Vulnerability</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 2.3: Evaluation results for LATAAX network. Maximum two-link failure vulnerability for LATAAX is \( |E| - 1 = 46 \). Topological limit for AFS and Vulnerability are 0.15 and 0.40 respectively.

Finally, utilizing MOPs can significantly improve network robustness as clearly shown in this table. With MOP selection, network vulnerability is significantly improved. Vulnerability of 9 to 13.15 links with CDP (depending on whether or not filtering is used) improves to 1 or less when MOP is used. It shows that for LATAAX, on average, out of the 46 possible second failures, the network is only vulnerable to 1 of them. Note that we can get the fraction of average connections that are left susceptible to a failure to the total number of connections that are affected by the failure by looking at the ratio between AFS and the average number of affected connections shown in Table 2.4 (note that recovery ratio is simply one minus this number).

Similarly, Table 2.4 shows all of the results from our simulations. The networks are arranged in a decreasing order of average node degree. The total number of connections represent the number of routed bidirectional connections. Average number of affected connections represent the number of connections that are af-
Table 2.4: Evaluation results for eight sample networks (Measurements after failure and recovery of a single link and before the second link failure). ARPANET’s Vul. and AFS can be reduced to 0 using the filtering algorithm because it is a 3-connected network.
fected by the initial failure (either primary is broken or the backup is no longer available), and therefore left susceptible to future failures. AFS and vulnerability measures before reconfiguration are shown only for the un-filtered scenario (AFS only varies by 2-5% and vulnerability by less than 1% and neither case is better than the other). The small initial difference seen here is due to the difference in the originally chosen routes, and have no real impact on the overall robustness of the network as we are considering reconfiguration techniques. The general pattern shown in the results with LATA X network holds true for all of the networks. As mentioned in the previous section, in general, networks with lower average node degree tend to benefit more from the filtering algorithm as the likelihood of initially selecting and provisioning poor path pairs increases with the decreasing average node degree. There is also a significant improvement in network robustness when using MOP selection over CDP selection. AFS improves by a factor of 3 or more depending on the network, and vulnerability improves by a factor of 7 to 20. The fundamental limits represent the maximum robustness (minimum AFS and minimum vulnerability) possible under the given topology, where it is limited only by physical disconnections. For all networks, except our example graph, Fun-Lim, utilizing the filtering algorithm along with protection reconfiguration under MOP selection allows the networks to achieve the maximum robustness.

Table 2.5 shows the reconfiguration costs. Initial capacity cost is simply the sum of all $\lambda$-channels used in provisioning a uniform full mesh demand with shared path protection. The difference in capacity cost between using and not using the robust path filtering algorithm is trivial. Neither case was better than the other in terms of this cost (difference was less than 2%), but, as discussed previously, filtering can significantly improve robustness. The small difference in cost is expected as different paths are selected in provisioning yielding different capacity sharing through online routing. There is about 20% difference between CDP and
Table 2.5: Reconfiguration cost results for eight sample networks. * symbol denotes ILP problems too large to run on our computers.

MOP selection in terms of reconfiguration cost because MOP selection assigns many more λ-channels for protection where CDP does not (because no protection paths can be found using CDP selection). The difference in capacity cost between filtered and not filtered is small, and are mostly due to online provisioning, which selects paths and computes reserve capacity sharing in a greedy manner (as connections arrive).

Optimal costs for reconfiguration with MOP were computed using CPLEX software with our ILP formulations for NJLATA, SubNJ, NSFNET and FunLim networks, and are shown at the bottom row of the table. The results from other networks are not available because the ILP problems are too large to run in a reasonable space and time run on the machines available to us. Finding the optimal reconfiguration cost, however, is not the focus of this work. The goal was to study the robustness of the networks. Obviously, using the simple greedy heuristic we used to assign reconfiguration capacity is one of the quickest ways to reconfigure the network, and running a full optimization would probably be the slowest method for reconfiguration. It is also important to note that, in practice, with a large traffic load or as the traffic load increases, full optimization process becomes less and less attractive as the network is left vulnerable during the time

<table>
<thead>
<tr>
<th>Network</th>
<th>NJ</th>
<th>COST</th>
<th>NATIONAL</th>
<th>LATAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Capacity Cost</td>
<td>314</td>
<td>1190</td>
<td>2552</td>
<td>4034</td>
</tr>
<tr>
<td>Not CDP Avg. Rcnf. Cost</td>
<td>29.3</td>
<td>82.6</td>
<td>178.1</td>
<td>377.1</td>
</tr>
<tr>
<td>Filtered MOP Avg. Rcnf. Cost</td>
<td>34.3</td>
<td>90.8</td>
<td>221.5</td>
<td>456.4</td>
</tr>
<tr>
<td>Filtered CDP Avg. Rcnf. Cost</td>
<td>29.6</td>
<td>76.2</td>
<td>167.6</td>
<td>381.7</td>
</tr>
<tr>
<td>Filtered MOP Avg. Rcnf. Cost</td>
<td>34.0</td>
<td>90.0</td>
<td>208.4</td>
<td>456.8</td>
</tr>
<tr>
<td>Filtered, MOP, Optimal Rcnfg. Cost</td>
<td>11.4</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network</th>
<th>ARPA</th>
<th>SubNJ</th>
<th>NSFNET</th>
<th>FunLim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Capacity Cost</td>
<td>1644</td>
<td>110</td>
<td>588</td>
<td>80</td>
</tr>
<tr>
<td>Not CDP Avg. Rcnf. Cost</td>
<td>246.9</td>
<td>14.9</td>
<td>104.6</td>
<td>15.5</td>
</tr>
<tr>
<td>Filtered MOP Avg. Rcnf. Cost</td>
<td>270.8</td>
<td>19.1</td>
<td>114.9</td>
<td>21.8</td>
</tr>
<tr>
<td>Filtered CDP Avg. Rcnf. Cost</td>
<td>250.9</td>
<td>14.9</td>
<td>99.6</td>
<td>17.5</td>
</tr>
<tr>
<td>Filtered MOP Avg. Rcnf. Cost</td>
<td>266.0</td>
<td>19.1</td>
<td>107.9</td>
<td>23.8</td>
</tr>
<tr>
<td>Filtered, MOP, Optimal Rcnfg. Cost</td>
<td>*</td>
<td>7.2</td>
<td>42.8</td>
<td>12.0</td>
</tr>
</tbody>
</table>
it takes to compute the optimal answer and follow up with reconfiguration. One simple approach is to quickly reconfigure the network using a greedy solution, and rearranging it once a better solution has been computed. It may also be possible to find a better heuristic in the problem space of cost versus computation time, but is outside the scope of this work and is left for future work.

2.4 Summary

In order to further guarantee reliable network services for increasing communications demands, we must understand the impact of different failure scenarios and provide effective ways to address the challenges that arise under different failure models. Graceful degradation is a key issue that must be considered in designing more robust and dependable future networks. To this end, multiple sequential failure survivability is an important measure of a network’s ability to operate effectively under failures.

Pre-planning multiple recovery routes and reserving enough wavelengths to allow a network to recover from multiple failures can be expensive in terms of protection capacity. In contrast, dynamic protection reconfiguration can allow a network to achieve high survivability under multiple failures while utilizing little additional capacity.

Our experiments showed that a large number of choices of working and protection path pairs limit the network from achieving optimal robustness from multiple failures. The algorithm we presented efficiently filters out these poor choices. Using our MOP selection algorithm, after using the filtering algorithm, allows the network to achieve optimal survivability under sequential failures.

On several representative networks, dynamic protection reconfiguration (with robust path filtering and MOP selection) required little additional capacity for
shared path protection while allowing networks to operate with the maximum multiple failure survivability for a given network topology, where only disconnections (topological separation) can leave lightpaths unrecoverable.
CHAPTER 3

Reliability-Differentiated Management

As demand increases for more robust and fluid communications to support our growing reliance on rapid access to information, the need for efficient and reliable networks becomes critical. The use of WDM technology in the backbone networks has enabled us to meet these demands by taking advantage of the huge capacity of optical fibers. Numerous protection schemes exist for these networks, but in practice most networks use only one or two such schemes, roughly classifying customers into those that need robust connectivity and those that do not. In this work, we examine the potential benefits of using a broader system of protection classifications to support data traffic and present a novel approach to optimization across classes that reduces the protection capacity necessary to support a given traffic load.

Most WDM backbones still carry primarily SONET (Synchronous Optical NETwork) streams, which in turn consist mostly of virtual ATM (Asynchronous Transfer Mode) circuits. IP (Internet Protocol) packets are then layered atop ATM, with virtual circuits providing the links between routers. However, as projected in [49], data communication volume has grown exponentially, while voice has grown only linearly. Currently, voice transmissions accounts for only a tiny fraction of total traffic, making the use of protocols designed to carry such traffic questionable.

SONET and ATM were both designed more than a decade ago by the telephony industry at a time when data traffic was essentially irrelevant in the wide area.
While they are both mature, well-established and well-tested protocols, they do not necessarily do a good job in addressing the needs of data traffic. One issue in particular is the inclusion of recovery functionality at all four layers mentioned, leading to inefficient use of physical resources as well as complex synchronization schemes to avoid interference between layers when a problem occurs.

Many researchers have thus begun to investigate the possibility of coupling the IP layer more closely to the WDM layer, removing most of the replicated functionality in SONET and ATM and moving the rest into IP, WDM, or a slim layer between the two [50]. If the layers are reorganized, the proper layer for protection functionality is unclear. These issues are currently addressed in markedly different ways in the two layers. Restoration time has long been considered an aspect of quality-of-service (QoS) in many circuit-switched networks like ATM [51, 52]. WDM protection schemes offer fast restoration, often on the order of the 60-millisecond restoration requirement imposed for SONET self-healing rings. In sharp contrast, recovery through Internet routing protocols, whether within an Autonomous System (AS) using Open Shortest Path First (OSPF) or between them using BGP-4 [53], can currently take minutes [54, 55]. Some claim that these long times are not fundamental to the protocols themselves, but in practice, security concerns with automatic routing updates have dramatically slowed the propagation of failure information with BGP-4, in which information is usually only forwarded to neighbors every 30 seconds [53].

We believe that protection functionality must be supported in both layers. WDM schemes that support restoration over several autonomous, independently-managed domains have yet to be developed, and are unlikely to be simple. Such recovery must occur within the IP framework. When possible, however, recovery should be fast to support applications that need high availability, such as air traffic control, remote surgery, and certain types of transactions. Protection at
the physical layer must thus also be made available, and customers allowed to
differentiate themselves according to their needs. As with most optimization
problems, relaxing constraints by allowing additional protection options reduces
the protection capacity requirements for a WDM network. Schemes in which IP
controls nearly all WDM-layer functionality [56] may be feasible, but a diverse set
of protection schemes is attractive.

A WDM network that supports several compatible protection schemes also
offers opportunities to optimize across connections using different schemes. In
addition to exploring the benefits of increased protection service differentiation,
this work describes an optimization for networks that offer both dedicated (one-
for-one, or 1:1) and shared (one-for-N, or 1:N) protection that allows capacity costs
to be reduced by as much as 15% when only these two schemes are supported,
and by 5-10% in a network with more protection schemes.

The remainder of this chapter is organized as follows. In Section 3.1, we
describe related background material in more detail. Section 3.2 outlines our
approach to protection-differentiated QoS and introduces our protection classi-
fications. Section 3.3 describes our methodology for evaluating the benefits of
differentiation and introduces an interesting optimization for 1:1/1:N protection.
Section 3.4 gives our results and a discussion of their meaning. Finally, a summary
is provided in Section 3.5.

3.1 Protection-based Service Differentiation

3.1.1 QoS Under WDM Networks

The idea of supporting protection differentiation in optics is not new, but neither
has it been thoroughly explored. Early work in this area [57, 58] primarily ad-
dressed issues of physical signal quality and blocking probability. More recently, a study proposed leveraging the emerging Multiprotocol Label Switching (MPLS) standards, which support the identification of the customer or group of customers behind a particular packet in a traffic flow through the use of labels within headers [59]. In coordination with the optics, the MPLS flow classification can then include a resilience class. This study [59] fairly clearly demonstrates the benefits of supporting multiple resilience classes for reducing protection capacity, but assumes that all unprotected traffic can be preempted in the event of a failure and performs off-line routing and optimization. We split unprotected traffic into preemptable and non-preemptable classes, as we believe that the increased vulnerability due to preemption will be unattractive to many customers. In IP/MPLS over WDM networks, many paths in the optical layer will be provisioned without any protection, therefore, preempting these traffic may have undesirable effects on the upper layer protocols (IP/MPLS, TCP etc.). In addition, off-line optimization is used in [59]. As both the size and the complexity of networks increases, especially in IP/MPLS over WDM networks, dynamic routing becomes more attractive than static routing as lightpaths will be required to be setup and torn down dynamically to meet the communication demands. We therefore perform online routing which does not allow off-line optimization that can aid in reducing cost in terms of capacity usage. We also present results on the average number of connections broken by a failure and the percentage of traffic that were protected for free (with zero capacity cost), thus providing more insight into these tradeoffs.

3.1.2 Survivability

Failures in optical networks result in loss of enormous data and revenue. Some of these failures include channel failures, link failures and failures of optical cross-
connects (OXC). Channel failures caused by card failures at a port of an optical switch are the most common type of failures in optical networks. Links failures (fiber cuts caused by wayward backhoes, amplifier failures etc.) are also common, and can result in failures of all the channels that are carried on the fiber. Node (OXC) failures are less common, but can cause failures of all the links that are adjacent to the node.

Protection and Restoration are the two main approaches that address failures in optical networks [60, 61]. Restoration addresses failures by locating free λ-channels for backup after a failure occurs. Protection preplans backup routes that are used in the event of a failure. Protection and restoration offer a tradeoff between the speed of recovery and efficiency in terms of the use of spare capacity [11, 12]. However, protection can be implemented in a capacity efficient manner [15, 16, 17, 18, 19] and can offer much faster recovery than restoration with the absence of the signaling delay needed for dynamic route discovery [20, 21, 22]. Restoration schemes find a recovery route dynamically, which takes about 2 seconds, whereas protection schemes can achieve complete recovery in the order of tens of milliseconds [23]. We therefore focus on protection, and for the rest of this chapter, we use the terms restoration and protection interchangeably to mean protection as defined above.

There are two types of protection: local (link or node) protection and path protection. Path protection requires the knowledge of the whole path and selection of a backup path that is shared risk group (SRG) disjoint from the primary.
In 1+1 protection, traffic is sent out over both paths and the receiving node simply switches to the backup stream in the event of a failure [23]. 1+1 protection offers very fast recovery with little data loss because no signaling is required between the source and the destination nodes, but is inefficient in terms of capacity requirements. 1:1 protection is same as 1+1 except the data stream is not actively sent out, but switched after a failure. In shared path protection schemes, the end nodes of a lightpath signal the intermediate nodes to establish the backup route. Capacity reserved for backup can be shared among different connections that do not share same SRGs, or can also be used to carry low priority (unprotected) traffic, which is preempted in the event of a failure. The signaling and configuration of the intermediate PXCs render shared mesh protection slow compared to 1+1/1:1 protection. In link protection, nodes that are adjacent to the failure initiate recovery by reserving spare capacity and signaling and configuring the intermediate nodes after a failure in a manner akin to path protection. However, recovery of failures usually involves the use of more local resources compared to path protection. Recovery is usually faster because it is initiated by the end nodes of the failed link (vs. path protection), but link protection is less efficient in terms of spare capacity usage [16, 18].
3.2 Protection Differentiated QoS

Different network clients and applications have different survivability needs ranging from mission critical applications requiring immediate recovery with minimized data loss to lower-end user traffic with no survivability needs. Different protection algorithms offer different protection capabilities such as speed of recovery, data loss, provisioning costs and management overhead. Utilizing link protection and dynamic restoration for different classes of traffic can provide sufficient differentiation among traffic classes with different survivability needs, but at the cost of having two different protocols to operate and manage. In order to reduce management overhead, we choose to utilize a single class of protection algorithms. For this reason, we focus on path protection to meet our goal to lower operation costs through protection differentiated QoS. In this section we propose a five classification schemes and discuss the details of each protection class.

3.2.1 Protection Classes

Table 3.1 shows the proposed classification scheme for protection based QoS support in optical networks. We next briefly explain each class differentiated by protection requirements.

Priority Class (Class A)

Mission critical traffic that require high availability, low loss service can utilize lightpaths of this class. Dedicated path protection (1:1 or 1+1) is used for this class of service and achieves the highest level of protection. Recovery of a link failure takes about 20ms for 1+1 or 40ms for 1:1. Up to about 20ms (failure detection and switching time at the end nodes, and possibly propagation delay) of data is lost after the failure. Protection resources are pre-allocated and the
recovery paths are preconfigured (paths are computed and the switches along the paths are pre-set). Less data is lost when 1+1 is used, but 1+1 is more expensive operate compared to 1:1 because traffic needs to be actively duplicated and sent out over two live paths in the network. When 1:1 protection is used, protection paths can be used to carry the pre-emptable class traffic to reduce capacity cost.

Protected Class (Class B)

Service classes with a lower level of protection requirement can be assigned to Class B. Shared path protection (1:N) is used for this class. Recovery paths are computed, but the switches along the paths are not preconfigured. This flexibility allows sharing of protection resource among different lightpaths and reduces capacity cost. Recovery takes about 90ms to complete with 50 to 90ms of data loss.

Reroutable Class (Class C)

Reroutable traffic are given shortest path working paths and have no protection resource allocated for use. However, best effort rerouting may be done after a failure to recover some of the Class C traffic. Rerouting is done using the unused protection resources allocated for Class A and B after the Class A and B traffic are fully recovered. The average number of Class C traffic that cannot be rerouted after a failure is given in Section . Network service providers may reserve additional capacity to increase the recovery ratio. Shortest paths are assigned for this type of traffic to reduce total capacity cost. Rerouting can begin immediately after a failure and can take up to several seconds.
Unprotected Class (Class D)

Class D traffic are also assigned shortest available paths in the network to reduce capacity cost. They have no protection from failures and cannot be rerouted. Data is lost until a physical repair is made.

Pre-emptable Class (Class E)

Pre-emptable class traffic are the cheapest to provision. Routing can take advantage of resources that are already provisioned for protection of either Class A or B traffic to reduce capacity cost. Furthermore, Unprotected traffic can be pre-empted to make room for rerouting class C traffic in case of a failure. Generally, data is lost until a physical repair is made, but more data can be lost if lightpaths were pre-empted to make room for Class A or B’s recovery. Lightpaths that were pre-empted are brought back only after having the Class A or B traffic restored to their original working paths.

3.2.2 Classification Scheme

Table 3.2 shows 7 different protection based differentiation schemes that we evaluate. S1 and S2 represent 5 class differentiation scheme we propose for QoS routing at the optical layer. S1 improves capacity performance over S2 by using a novel sharing optimization explained in the next section. S3 and S4 consist of two classes of traffic differentiated by whether or not protection is provided. They only differ in the choice of protection algorithm used for the protected class traffic. S5–S7 are based on single class traffic. In S5, all lightpaths are unprotected. In S6 and S7, all lightpaths are protected. Like S3 and S4, S6 and S7 differ only by the choice of protection algorithm used.
3.3 Capacity Assignment

An important motivation for having protection based QoS is to reduce network operation costs. An efficient capacity assignment scheme for protection based QoS is needed as the classification and the choice of protection services directly affect cost in terms of provisioned network capacity.

3.3.1 Routing and Wavelength Assignment

We assume uniform traffic demands which can effectively aid in capturing the different characteristics of the classification schemes. In the simulations, we perform dynamic on-line provisioning with uniformlly distributed full-mesh traffic demands scaled by a factor of 10. Dynamic provisioning means that we have no knowledge of future demands, and cannot reroute existing connections on the network to optimize provisioning upon receipt of a new request. Each request is assumed to be a bidirectional connection with a uniformly distributed demand of 1 lightpath between each source and destination. Table 3.2 shows the traffic ratios between each class of traffic for the different classification schemes where 1 equals a uniformly distributed demand of full-mesh, \((N\times(N-1))/2\), bidirectional requests. Traffic demands are routed in random order to simulate an on-line provisioning process. Although, in practice, the demands may not be uniformly distributed among different requests, we believe that studying uniformly distributed traffic demands is sufficient in that it shows the characteristics of different protection schemes for comparison purposes. We assume that each \(\lambda\)-channel has a cost of 1 in terms of calculating capacity. The total cost of capacity is therefore the sum of the overall of working paths and the total number of the reserved protection \(\lambda\)-channels.

For both Class A and Class B with 1+1/1:1 and 1:N protection, we utilize a joint path selection method similar to the one used in [15]. The working and
protection paths are selected together to minimize the capacity cost. We always route classes C and D using shortest paths. If Class E exists in the classification scheme, then routing depends on whether or not protection resources are reserved in the network. Class E lightpath are routed over an existing dedicated protection paths with the same source and destination. If protection resources are allocated to protect Class B, we find paths such that the cost is minimized via sharing with Class B’s protection resources. If no sharing is possible, the algorithm automatically will choose shortest paths.

3.3.2 Sharing Optimization

The key to our optimization algorithm is the sharing of protection resources between two different protection differentiated classes utilizing dedicated path protection (Class A, 1:1/1+1) and shared path protection (Class B, 1:N). Preconfiguration of switches is the main difference between 1:1/1+1 and 1:N protection. Since switches are not preconfigured, 1:N algorithm can allow sharing between multiple protection paths as long as their working paths do not share a common failure mode. Paths protected by the 1:1/1+1 scheme cannot share resources with other 1:1/1+1 schemes because the switches must be preconfigured in order to provide rapid recovery.

We assign protection resources such that resources can be shared between protection paths if their working paths do not share common failure modes. 1:N, Class B, protection paths can share resources with any other protection path(s). In the optimized version of the sharing algorithm, a single Class A lightpath can share a protection channel with any number Class B lightpaths. Switches are then preconfigured to support recovery of the Class A lightpath, and when needed, reconfigured to support recovery of Class B lightpaths.
Figure 3.1: The National network.

<table>
<thead>
<tr>
<th>Network</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
</tr>
</thead>
<tbody>
<tr>
<td>National</td>
<td>17414 (1.00)</td>
<td>18788 (1.079)</td>
<td>18862 (1.083)</td>
<td>22894 (1.315)</td>
<td>16000 (0.919)</td>
<td>25380 (1.457)</td>
<td>39000 (2.340)</td>
</tr>
<tr>
<td>Arpanet</td>
<td>11314 (1.00)</td>
<td>12270 (1.084)</td>
<td>12252 (1.083)</td>
<td>15164 (1.340)</td>
<td>10460 (0.925)</td>
<td>16334 (1.444)</td>
<td>26180 (2.314)</td>
</tr>
<tr>
<td>Cost 239</td>
<td>8286 (1.00)</td>
<td>8964 (1.082)</td>
<td>8944 (1.080)</td>
<td>10906 (1.316)</td>
<td>7660 (0.924)</td>
<td>11818 (1.426)</td>
<td>18480 (2.230)</td>
</tr>
<tr>
<td>Lata X</td>
<td>27004 (1.00)</td>
<td>29038 (1.075)</td>
<td>29360 (1.087)</td>
<td>35430 (1.312)</td>
<td>24840 (0.920)</td>
<td>39810 (1.474)</td>
<td>60140 (2.227)</td>
</tr>
<tr>
<td>NJ Lata</td>
<td>2134 (1.00)</td>
<td>2260 (1.059)</td>
<td>2294 (1.074)</td>
<td>2682 (1.257)</td>
<td>1920 (0.900)</td>
<td>2930 (1.373)</td>
<td>4460 (2.089)</td>
</tr>
</tbody>
</table>

Table 3.3: Total capacity cost for different classification schemes normalized to class S1.

### 3.4 Evaluation

Figure 3.3 shows the results of on-line provisioning performed on the National network (US Backbone with 24 nodes and 44 links, shown in Figure 3.1) with the seven different classification schemes previously explained in section 3.B. Protection requirements for all Class A and B traffic can be met with 8.8%(S1) and 17.4%(S2) additional capacity compared to S5, which employs all unprotected traffic. S6 and S7 requires over 148% additional capacity compared to S1. Note that Class D traffic can be converted to class C traffic on S3 and S4 at no additional cost in terms of capacity.
It is interesting to note that protection capacity on S1 is very close to protection capacity on S3. S3 consists of all Class B traffic, and therefore the authors are showing that the sharing optimization allows enough sharing of protection resources between Class A and Class B traffic that the efficiency is equivalent to using all 1:N protection. Figure 3.4 more directly shows the benefit of the optimization. The total traffic shown on Figure 3.4 is consistent with the demand used for results on Figure 3.3. The ratio between Class A and Class B is varied from 0 to 100 to show the optimization. At 33.3%, pointed by the arrows, the overall capacity cost is improved by 7.9% as also shown in Table 3.3. We also measured the on-line provisioning cost in terms of capacity using the classification scheme provided in [59]. All demands can be provisioned with an addition of less than 1% capacity over S5. The improvement comes from assuming that all unprotected traffic belong Class E (preemptable). Grouping all unprotected traffic to Class E is not
Figure 3.3: On-line capacity provisioning results on the National network. attractive because lightpaths provisioned under Class E are susceptible to failures of other lightpaths.

We also simulated on-line provisioning using four other sample networks shown in Figure 3.2. Table 3.3 shows the total capacity results for different classification under each network. Results show that the benefits of the differentiation via protection classification is consistent for the five sample networks used where S1 provides 7.4 to 8.4 percent improvement in capacity cost over S3.

Table 3.4 shows the average failure count of each class of traffic under different protection differentiated classifications. The average number of lightpaths that are affected by a single link failure (average link load) for each protection class is also shown. Since recovery for Class C utilizes rerouting over existing protection capacity, reducing capacity through sharing optimization reduces the available resources for Class C. For S1, 91.0 out of 137.9 Class C lightpaths cannot be rerouted whereas for S2, 45.3 out of 135.6 Class C lightpaths are left unrestored.
Figure 3.4: Capacity for Class A and Class B traffic with varying percentage of Class A traffic for National. Arrows point to data at 1:2 ratio corresponding to data shown in Figure 3.3.

Table 3.4: Avg # of failed lightpaths / Avg. # of lightpaths affected by a link failure (avg. link load).
3.5 Summary

A protection differentiated classification scheme based on five protection classes was proposed. We also introduced a novel sharing optimization method that allows sharing of protection capacity between two different classes of traffic. We showed that using protection based classification can reduce network capacity cost by up to 130% on average over five sample networks. Results showed that about 8% additional capacity cost can be reduced by using our sharing optimization under protection differentiated classification.
CHAPTER 4

Streams: Rapid and Efficient Recovery Algorithm

All-optical networks (AON) offer several critical advantages over optically opaque networks. Faster switching can be achieved with the absence of electronic and photonic processing delays that can act as a bottleneck on the total transmission time. At the same time, the absence of high-speed electronics may offer a significant reduction in equipment costs. In addition, unlike opaque networks, TONs can handle signals with different data rates, protocols, and formats, making it more suitable for supporting future changes. On the other hand, they suffer from limited functionality in wavelength conversion, signaling capabilities and detailed performance monitoring [23, 62]. These limitations reduce the efficiency of many of the recovery schemes studied in the literature and make implementation more challenging [63]. These issues, coupled with the fact that the impact of failures in TONs is exacerbated by their high traffic volumes, necessitate a better understanding of the tradeoffs between different survivability schemes. In considering survivability options, understanding the tradeoffs between recovery speed, data loss, capacity requirements, and the implementation overhead of different protection schemes is imperative. Our goal then is to provide rapid failure recovery in TONs in an efficient manner, and reduce service disruptions and loss of data. In this work, we evaluate different protection algorithms and study their tradeoffs. We also distinguish the difference between recovery speed and data loss and show that backup path lengths have minimal impact on data loss.
4.1 Differentiation of Recovery Time and Data Loss

Although the primary metrics for evaluation of recovery algorithms have been protection capacity requirements and speed of recovery, most studies in the literature do not distinguish between overall recovery time and the actual period of data loss [9, 14]. Data loss occurs until the traffic is diverted to the backup path, but recovery is complete until the backup traffic reaches the destination.

The time between a failure and initiation of recovery differs among the schemes discussed above. For link-based protection, failure detection requires approximately one link propagation delay, whereas path-based protection only reacts to a failure after a propagation delay on about half of the whole path on average. Since data is lost until backup traffic is sent out over the backup path, the total period of data loss, therefore, includes both the time for detecting a failure and the time it takes switch over to the backup path. As a result, more data is lost with path protection relative to link protection, and more data is lost with shared backup schemes that require signaling and switching of intermediate nodes after the failure. With protection schemes such as generalized loopback and flooding-based mesh restoration, even though the backup capacity is shared, flooding is initiated immediately upon detection of a failure preventing further loss of data. The Streams falls under the same category and this distinction is discussed in detail for the context of our experiment in section 4.3.2.

4.2 Streams

We present a protection algorithm called Streams, which allows for rapid recovery from all single link or single node failures, and is comparable to dedicated path protection in terms of recovery speed while utilizing 10-35% less total capacity
with non-dynamic traffic or significantly increasing the number of connections a network can support at a given blocking rate with dynamic traffic loads. The Streams algorithm employs the same recovery protocol used for 1:1 (one-for-one) dedicated path protection (DPP). Streams is powerful because it can also coexist with other algorithms in the same network as it does not require additional equipment support. It can readily be implemented on current networks that utilize DPP or shared path protection (SPP). The Streams algorithm provides recovery speed comparable to DPP or a ring-based protection scheme, but is more efficient in terms of capacity and blocking probability. Streams can achieve capacity efficiency closer to SPP. SPP and Streams offer an interesting tradeoff between capacity and backup path length. By adjusting the degree to which backup path lengths can be extended, a network can be operated at different points in the space of capacity versus path length expansion.

The concept utilized by the Streams algorithm was independently developed by the authors and introduced in the context of all-optical networks in [64] and by another group in the context of OEO networks in [65]. Both papers present simple heuristics for link failure protection under non-dynamic routing scenarios. In this work, we provide a much more detailed study of the tradeoffs between different well-known protection algorithms and introduce a simple heuristic for routing dynamic traffic for Streams and show the performance tradeoffs on several well-known networks. We also evaluate node failure scenarios in detail. We present the relative performances in terms of capacity and backup path length expansion as well as providing results that show different operating points at which networks can choose to utilize the different protection algorithms. We found several interesting phenomenas in the tradeoffs between different protection algorithms, traffic load and network topology. The details are provided in the results section. Finally, we also formulate an integer linear programming (ILP) solution for Streams and
evaluate capacity provisioning for static, offline traffic to show that the Streams protection scheme provides the same advantages for offline provisioning.

### 4.2.1 Streams Protection Scheme

![Figure 4.1: A) SPP setup B) Example of a Streams setup](image)

We now present the Streams algorithm, which can be applied to any two-node connected network (or two-edge connected network, if node failures are not considered). Streams can be thought of as a virtually shared-DPP algorithm. It is like DPP in the sense that all PXCs are preconfigured at the time a lightpath is provisioned, and in the event of a failure, backup traffic is simply sent over the pre-established backup path, termed a *stream*. Preconfiguration enables the PXCs to switch over to backup routes without performing any decision making in the event of a failure, and aids our goal for rapid recovery. All PXCs along a stream are preconfigured to simply forward the backup traffic along the reserved $\lambda$-channels in a specific ingress to egress port setting (identical to 1:1 DPP). The mapping from ingress ports to egress ports at intermediate PXCs is maintained in the PXC configurations themselves, and is updated when lightpaths are provisioned or torn down or in the event of a failure. Recovery with Streams is much faster than with protection algorithms that use soft-reserved backup capacity (such as SPP), as no signaling or configuration of intermediate PXCs is required after a failure. The key difference is that it allows sharing of a stream across different connections. Streams fall somewhere between DPP and SPP in terms of the
existing classification of survivability techniques. Each connection is protected by a backup path, which must lie entirely on a single stream. The stream may extend beyond the backup path in either direction. Backup paths that are on the same stream cannot diverge. The top example in Figure 4.1 illustrates an SPP-like solution, which allows diverging paths to share resources. In the example, two backup paths are sharing a wavelength on link \([A, B]\). If a failure of one of the two respective primaries fails, the end node for the failed primary will need to signal the nodes along its backup so that the PXC’s are properly configured to carry the backup traffic. In this example, without signaling, node \(B\) does not know where to forward the backup traffic, and for the same reason, it cannot be preconfigured as it can only forward data in one direction. Therefore, the backup paths may not share wavelengths.

In Streams, sharing backup wavelengths is only allowed between backup paths that do not diverge. The example shown on the bottom of Figure 4.1 is a possible setup under Streams protection. Here, three backup paths are shown with wavelengths shared on links \([B, C]\) and \([C, D]\). Detailed descriptions of how the Streams protection scheme works are presented in the following sections. We show that the constrained form of sharing in Streams still allows a significant reduction in capacity. Detailed results are presented in later sections.

In short, what makes Streams attractive is that the recovery protocol is extremely simple, and requires no additional hardware support (relate to DPP or SPP). There is a small overhead in route computation, but the recovery protocol/implementation remains identical to 1:1 DPP. The remainder of this section describes the failure detection and recovery process followed by our simple online and dynamic Streams algorithms.
4.2.2 Failure Recovery Process

The failure detection-recovery initiation process is simple: for a given channel, the PXC at the end of the lightpath detects a failure by monitoring the signal quality on the channel (many metrics are possible [66]). For simplicity, we assume bidirectional connections, which allows the receiver to initiate recovery as soon as it detects a failure. When a failure is detected, the end-nodes immediately redirect traffic to the assigned stream (preconfigured backup path). At the same time, the two nodes reconfigure themselves to begin listening to the preassigned ingress port for backup traffic. This reconfiguration step does not add to the recovery delay as it is performed simultaneously with redirection of the traffic.
A node may also be both an end node (source or/and destination) for a connection protected by a stream and an intermediate node for the same stream as shown in Figure 4.2 (Node $B$ is the end node for lightpath $P_1$ and is also an intermediate node along the stream protecting $P_1$ and $P_2$). In this case, after the connection is recovered, the configuration is fixed preventing possible reconfiguration and signal collisions in the event of additional failures. The following example illustrates the Streams setup and the recovery process.

In Figure 4.2, two connections $P_1(D - A - B)$ and $P_2(E - F - C)$ are protected by the stream $D - E - B - C$. On the figure, only single directions are shown for presentation purposes. The intermediate nodes $E$ and $B$ along the stream are preconfigured (similar to DPP) to simply forward traffic from $D$ to $B$ and $E$ to $C$ respectively. Suppose either link $[D, A]$ or $[A, B]$ fails and disrupts $P_1$. Upon detecting a failure on $P_1$, the source node, $D$, redirects traffic onto the protection stream by simply sending out the traffic towards $E$. Intermediate nodes on the protection stream—$E$ and $B$—are configured to simply forward the traffic along the stream. $B$ (destination node), however, is also an end node for $P_1$, and therefore reconfigures itself to receive the traffic coming from $E$ to the drop port for $P_1$ instead of forwarding it to $C$. The role of $B$ and $D$ are simply switched for the traffic originating at $B$ and terminating at $D$. The recovery process for a failure on $P_2$ is also shown in the figure. If there is a non-simultaneous multiple failure, say, without loss of generality, a failure on $P_1$ followed by a failure on $P_2$, it is clear that both connections cannot be recovered (the protection algorithm is designed to handle single failures only). Nodes along the stream that are being used for recovery will lock its configuration (until the connection is restored) so that it ignores future failures that may otherwise result in its usage. Therefore, $E$ knows that the stream is being used by $P_1$ and does not attempt to send the traffic for $P_2$ over the stream (prevents switching on $E$). Note that the actual
recovery process is identical to DPP, where post-failure switching takes place only at the end nodes.

It is important to note that the Streams algorithm does not limit the detection process to the signal monitoring approach described. Other approaches that allow quick detection and propagation of failure information to network nodes can also be used, allowing existing networks to readily adopt the Streams protection algorithm. For unidirectional connections, common recovery initiation techniques used by other algorithms, such as DPP, can be used. Most importantly, the detection time across the different protection schemes are the same as long as the same method is utilized.

4.2.3 Lightpath Provisioning

We assume that protection is not provided for source and destination node failures; if such failures are handled at all (for any survivability technique), they generally make use of redundant node hardware and redundant connections to the optical network, both of which are orthogonal to the network recovery algorithm.

![Figure 4.3: A working example of the Streams protection scheme.](image)
The primary and backup paths for a connection must obey two constraints. First, both paths must obey the wavelength continuity constraint, i.e., the same wavelength must be used along the entire path. A pair of primary/backup paths can, however, use different wavelengths [67]. Second, for link failure protection, primary and backup path pairs must be link disjoint, and node disjoint for node failure protection.

A simple example of how Streams protection is set up is shown in Figure 4.3. In the figure, there are a total of six connections provisioned on the network with a single stream providing protection for all six connections. Solid lines with arrows represent the stream with its preconfigured forwarding directions. The dashed lines represent primary paths that are protected by this stream.

This example illustrates two subtle details about the Streams algorithm. First, a stream may be shared across multiple connections that have common failure modes. In other words, multiple connections that share a link (or a node) may use the same stream for protection if these connections utilize different parts of the stream for protection. In the example, two connections are assigned on link [3-8], and a failure on this link will simultaneously break the two connections. It is easy to see that the two connections use different parts of the stream. Second, a primary path can share common failure modes with the stream providing protection as long as the links or nodes they share are not needed for recovery. In the example, the connection 2 to 8 shares a link ([4-8]) with the protection stream. Again, it is clear from the example that the failure of link [4-8] does not affect recovery of connection 2 to 8.

In the next section, we utilize simple heuristics to set up Streams, and evaluate the Streams algorithm using two online routing scenarios—non-dynamic provisioning and dynamic provisioning. For both scenarios, a greedy approach is used to perform a joint search for the best cost primary and backup path pairs for each
connection request, assuming no knowledge of future requests. The details are discussed in the following sections.

Non-dynamic Provisioning

For the non-dynamic provisioning scenario, the objective is to find the optimal primary and backup path pairs as connection requests arrive. An outline of three functions are shown below. First, route establishes a connection between a source (src) and destination (dst) pair. Shortest primary paths are used, and backup path lengths depend on the parameter $ex_{\text{hop}}$, which represents the number of extra hops allowed for backup paths. Second, evaluate determines the cost of a candidate primary and backup path pair and checks for its validity. It also checks to see if an existing stream can be used or extended to save capacity. Finally, update_network allocates resources for the new connection and updates the network status. The outline shown allocates connections such that they can be recovered from all single link failures. Performing intersection on the PSB set for nodes computes backup paths that cover all single node failures.
- Note: A path is treated as an ordered set of links.
- \( P(\text{src}, \text{dst}) \) – Set of shortest primary paths for the node pair (\( \text{src}, \text{dst} \))
  - This set is pre-computed.
- \( B(p_i, h) \) – Ordered set of backup paths corresponding to \( p_i \in P \)
  - sorted by length in ascending order from shortest length paths to
    (shortest length + \( h \)) hop paths [pre-computed]
- \( S \) – Set of streams [initially empty]
- \( \lambda(s) \) – Wavelength used by stream \( s \in S \) [initially empty]
- \( Free(\lambda) \) – Set of free \( \lambda \)-channels on wavelength \( \lambda \)
  - (subset of the set of links). [available link capacity]
- \( PSB(s, l) \) – Allocated primaries that use the same backup resource(s) upon failure.
  - link \( l \) on stream \( s \) (using \( \lambda(s) \)) is used for recovery if a link in
    this set fails. In other words, this set contains all primary paths that are
    protected by a part of \( s \), including \( l \).
- compatible \((s, b)\) – Checks for compatibility between stream \( s \) and backup \( b \)
  - on \( \lambda(s) \). Specifically, this method checks for possible splits/merges
    that may arise as a result of adding \( b \) to \( s \).
- find_first_fit_wavelength \((a)\) – Finds the lowest wavelength \( w \) where
  path \( a \) fits. For some existing \( w \) if \( a \subset Free(w) \) then \( \lambda(s) \)
  - is set to \( w \). A new wavelength is allocated if \( a \) does not fit in any of the
    existing wavelengths.

```plaintext
evaluate \((p, b)\) { //get compatible stream & cost
  mincost ← \(|\text{links}| + 1
  for all \( s_i \in S \) { //find least cost stream
    if (compatible \((s_i, b)\)) {
      cost ← 0
      valid ← true
      for all links \( l_j \in b \) {
        if ((\( p \cap PSB(s_i, l_j) \)) = \text{not empty})
          valid ← false
        if (\( l_j \notin s_i \))
          cost ← cost + 1 }
      if (valid AND cost < mincost) {
        mincost ← cost
        stream ← s_i
      }
    }
  }
  if (mincost = \(|\text{links}| + 1) \{ //make new stream
    stream ← b
    mincost ← length(b)
  }
  return \([\text{stream, mincost}]\)
} 73
```
Dynamic Provisioning

The dynamic provisioning scenario differs from the non-dynamic provisioning presented in the previous section in that the number of wavelengths on each link is limited, and therefore the objective is to achieve higher utilization by accepting as many calls as possible. We assume dynamic connection arrival and departure, and utilize a greedy selection of minimum cost primary and backup path pairs. If the available resources are insufficient to provision a call within its hop limit, the call is blocked. Instead of simply imposing a maximum path length, we look at bounding primary and backup paths separately based on the shortest path lengths.
(for the network graph) for each connection. This method is similar to how the backup path lengths were limited in the non-dynamic provisioning scenario. The advantage of using this method is that shorter paths are not favored in searching for alternate paths during route computation.

The algorithm we used for our study is similar to the heuristic described in the previous section. There are three key differences. First, the number of wavelengths on each link is limited. Second, because the number of wavelengths are limited, non-shortest primary paths are considered, and the number of additional hops allowed is used as a parameter. Finally, unnecessary wavelengths are freed upon connection departures.

A rough outline is shown below. When a call arrives, \textit{connection\_request} attempts to establish a primary and backup (on a stream) paths. The call is blocked if the paths cannot be provisioned within its hop limit. This method checks for compatible backup paths by using \textit{evaluate}. If the connection can be provisioned, \textit{update\_network} updates the network resource information as well as the stream information by allocating wavelengths to the primary and backup paths. Upon departure of a call, \textit{remove\_connection} updates the network resources and the stream. This step involves freeing wavelengths from the primary path and parts of the stream (if possible).
Note: A path is treated as an ordered set of links.

- c - Connection information that includes primary path, primary wavelength, backup path, and the stream it utilizes
- \( P(src, dst, h) \) - Ordered set of primary paths for the node pair \((src, dst)\) sorted by length in ascending order from shortest length paths to (shortest length + \( h \)) hop paths [pre-computed]
- \( B(p_i, h) \) - Ordered set of backup paths corresponding to \( p_i \in P \) sorted by length in ascending order from shortest length paths to (shortest length + \( h \)) hop paths [pre-computed]
- \( S \) - Set of streams [initially empty]
- \( s(c) \) - The stream used by connection \( c \) for protection [initially empty]
- \( L \) - Number of wavelengths per link [link capacity]
- \( \lambda(s) \) - Wavelength used by stream \( s \in S \) [initially empty]
- \( Free(w) \) - Set of free \( \lambda \)-channels on wavelength \( w \)
- \( PSB(s, l) \) - Allocated primaries that use the same backup resource(s) upon failure.

\[ \text{link } l \text{ on stream } s \text{ (using } \lambda(s) \text{) is used for recovery if a link in this set fails.} \]

In other words, this set contains all primary paths that are protected by a part of \( s \), including \( l \).

- compatible \((s, b)\) - Checks for compatibility between stream \( s \) and backup \( b \) on \( \lambda(s) \).

Specifically, this method checks for possible splits/merges that may arise as a result of adding \( b \) to \( s \).

- \( \text{find_first_fit_wavelength (a)} \) - Finds the lowest wavelength \( w < L \) where path \( a \) fits.

For some existing \( w \) if \( a \subset Free(w) \) then \( \lambda(s) \) is set to \( w \).

Returns -1 if a cannot be allocated (insufficient resources).

```c
// handle changes due to accepted connection requests
update_network (p, p_λ, b, s) {
    Free(p_λ) ← Free(p_λ) \ p
    new_links ← b \ s
    Free(\lambda(s)) ← Free(\lambda(s)) \ new_links
    \lambda(s) ← find_first_fit_wavelength (all_links)
    for all links \( l_i \in b \)
        PSB(s, l_i) ← PSB(s, l_i) \cup p
    s ← b \cup s
}
```
connection_request \((src, dst, pri_exHop, bck_exHop)\) \{  
\[\text{mincost} \leftarrow |\text{links}| + 1\]  
for all \(p_i \in P(src, dst, pri_exHop)\) \{ // consider all primaries  
for all \(b_j \in B(p_i, bck_exHop)\) \{ // and all backups  
if ((\(p_i\lambda \leftarrow \text{find\_first\_fit\_wavelength}(p_i)) > -1) \{  
[\text{stream}, \text{cost}] \leftarrow \text{evaluate}(p_i, b_j)  
if (\text{cost} < \text{mincost}) \{  
\text{mincost} \leftarrow \text{cost} + \text{length}(p_i)  
p \leftarrow p_i  
p\lambda \leftarrow p_i\lambda  
b \leftarrow b_j  
s \leftarrow \text{stream}  
\}\}\}\}  
if (\text{mincost} = |\text{links}| + 1)  
\text{block call}  
else \{  
\text{update\_network}(p, p\lambda, b, s)  
c \leftarrow [p, p\lambda, b, s]  
return \text{c}  
\}\}  
\}\}  
\}\}  
\}\}  
}  
//handle changes due to connection departures  
remove\_connection\(c\) \{  
[p, p\lambda, b, s] \leftarrow c  
Free(p\lambda) \leftarrow Free(p\lambda) \cup p  
for all links \(l_i \in b\) \{  
\text{PSB}(s, l_i) \leftarrow \text{PSB}(s, l_i) \setminus p  
if \text{PSB}(s, l_i) = \text{empty}  
\text{Free}(\lambda(s)) \leftarrow \text{Free}(\lambda(s)) \cup l_i  
\}\}
evaluate \((p, b)\) { //get compatible stream & cost
\[
\begin{align*}
\text{mincost} &\leftarrow |\text{links}| + 1 \\
\text{for all } s_i \in S \{ //\text{find least cost stream} \\
&\text{if (compatible } (s_i, b) \text{) } \\
&\quad \text{cost} \leftarrow 0 \\
&\quad \text{valid} \leftarrow \text{true} \\
&\quad \text{for all links } l_j \in b \{ \\
&\quad \quad \text{if } (((p \cap PSB(s_i, l_j)) = \text{not empty}) \\
&\quad \quad \quad \text{or } (l_j \not\in \text{Free}(\lambda(s_i)))) \\
&\quad \quad \quad \text{valid} \leftarrow \text{false} \\
&\quad \quad \text{if } (l_j \not\in s_i) \\
&\quad \quad \quad \text{cost} \leftarrow \text{cost} + 1 \} \\
&\quad \text{if } (\text{valid AND cost < mincost}) \{ \\
&\quad \quad \text{mincost} \leftarrow \text{cost} \\
&\quad \quad \text{stream} \leftarrow s_i \\
&\quad \}\} \\
\text{if } (\text{mincost} = |\text{links}| + 1) \{ //\text{make new stream} \\
&\quad \text{temp} \leftarrow \text{find_first_fit_wavelength } (b) \\
&\quad \text{if } (\text{temp} = -1) \\
&\quad \quad \text{block call} \\
&\quad \quad \text{stream} \leftarrow b \\
&\quad \quad \lambda(\text{stream}) \leftarrow \text{temp} \\
&\quad \quad \text{mincost} \leftarrow \text{length}(b) \} \\
&\text{return } [\text{stream, mincost}] \\
\end{align*}
\]

Figure 4.4: Networks used for evaluation.
4.3 Evaluation of Protection Schemes

First, we discuss two protection algorithms (dedicated and shared path protection) that are used in evaluating the Streams algorithm. Second, we outline the difference between recovery time and data loss for the different algorithms. We then present the simulation results from the online non-dynamic provisioning scenario and the results from the dynamic provisioning scenario, followed by our formulation of ILPs and results for offline static provisioning. The overall results follow our expectation in terms of the relative efficiencies in terms of capacity utilization. Streams fall somewhere between SPP and DPP, but closer to SPP. It is also important to understand that SPP will always be at least as efficient as Streams in terms of capacity utilization because every Streams solution can always be used for SPP, but not all SPP solutions are applicable to Streams as explained in the previous section. Computing each connection request in all of the online routing scenarios for all protection schemes we studied took less than a fraction of a second in the worst cases using a desktop equipped with AMD Athlon 1.5Ghz processor and 512MB of RAM.

Six different networks are used to evaluate the different protection algorithms; COST-small (11 nodes, 24 links), NJLATA (11,23), COST 239 (19,37) [68], NATIONAL (24,44), ARPANET (20,32), NSFNET (14,20). These networks are ordered by average node degrees from highest to lowest. The results provided in this section are also presented in the same order.

4.3.1 Dedicated and Shared Path Protection

DPP offers fast recovery with little or no data loss because no signaling is required between the source and the destination nodes after the failure. 1+1 DPP actively sends the backup traffic over the backup path, which is kept alive during the entire
lifetime of a connection. With 1:1 DPP, the backup path is only used when there is a failure, and therefore the backup path may be used to carry unprotected traffic. The unprotected traffic can be dropped in the event of a failure. This reuse makes 1:1 more efficient than 1+1 in terms of capacity; the tradeoff is the added delay in the recovery process of the 1:1 DPP. Once the traffic is placed on the 1:1 backup path, it takes about one propagation delay along the path to reach the destination node. DPP is routed using shortest primary path and the shortest corresponding backup path. DPP is easy to implement in AONs and often is used [63].

In SPP, channels are chosen in advance, but not preconfigured. Instead, the end nodes of a lightpath signal the intermediate nodes along its backup path to configure the switches after the failure occurs. Because the switches are not configured to forward traffic from/to specific nodes, capacity reserved for protection can be shared among different connections that do not share links in the primary path. Protection capacity can also be used to carry arbitrary unprotected traffic. The need to signal and configure intermediate switches renders SPP slow compared to DPP, but SPP requires significantly less capacity compared to DPP [19]. In simulating online routing, we use joint selection of link-disjoint primary and backup lightpaths to minimize the capacity cost in a manner similar to [15].

In practice, the signaling necessary to dynamically configure the intermediate switches after a failure can make implementation of SPP more complicated in AONs [63]. However, a simple implementation that leverages a bidirectional signaling wavelength per link may be utilized to solve the problem with the additional capacity of twice the number of links. Discussion of the details for such implementation is outside the scope of this work, and for the results shown in this work, this cost is ignored. Ignoring this detail results in a tiny reduction in the cost for SPP, and does not affect the relative tradeoffs highlighted in this work.
<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Failure detection</th>
<th>Total recovery</th>
<th>Data loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPP</td>
<td>10 ms</td>
<td>50 to 90 ms + $\delta$</td>
<td>50 to 90 ms</td>
</tr>
<tr>
<td>Streams</td>
<td>10 ms</td>
<td>20 ms + $\delta$</td>
<td>20 ms</td>
</tr>
<tr>
<td>DPP (1:1)</td>
<td>10 ms</td>
<td>20 ms + $\delta$</td>
<td>20 ms</td>
</tr>
<tr>
<td>DPP (1+1)</td>
<td>10 ms</td>
<td>20 ms + $\delta$</td>
<td>0 to 20 ms</td>
</tr>
</tbody>
</table>

Figure 4.5: Approximate data loss and recovery times. The $\delta$ terms denote the propagation delay on the backup path, which may vary slightly between the different protection schemes, but has no effect on data loss figures.

### 4.3.2 Recovery Time and Data Loss

We first discuss recovery times and data loss (the time during which the network loses data) for the different algorithms. Recovery time is an important aspect of designing and implementing a protection scheme, but data loss even more critical.

The numbers shown in Figure 4.5 are based on a few assumptions about the time required for basic operations presented in [9]. First, failure detection by the end nodes of a path takes about half of the propagation delay along the entire primary path. Since we are looking at mostly shortest paths, these numbers are the same for all three schemes we evaluate. For a reasonably large network, [9] assumed, for simplicity, that the total propagation delay on a path is about 20 ms on average. Again, this time does not vary much between the different protection schemes. Next, switching a single PXC takes about 10 ms, which is also consistent across the different schemes. Finally, signaling and configuration (usually uploading of maps) of intermediate PXCs take about 40 to 80 ms in SPP.

For the 1+1 case of DPP, recovery completes when the receiving node detects a failure and switches to the backup stream. With some effort, the backup signal can be delayed relative to the primary signal, allowing the receiver to avoid any data loss. With 1:1 DPP, there is some data loss until the source node switches to the backup path, and recovery is complete only after the data reach the destination node. Often, a very small bandwidth, usually on a separate, out-of-band channel,
is allocated for signaling as well as monitoring for failures. Forward and reverse alarm messages can be used in case of a failure occurs on just one direction of the bidirectional failure. If both directions fail, the end nodes can detect the failure and switch to the backup path. These assumptions, based on [9], are consistent across the different protection schemes and does not change the relative timing. Total data loss is thus roughly 20 ms with this approach, and recovery time is longer by one propagation delay on the backup path. Since the average backup path lengths may vary between different protection algorithms, we use the average hop count. Actual distances may provide exact timing figures, but it is sufficient enough to use hop counts to see the relative timing effects. It is important to note that the backup path lengths have no effect on data loss. Streams has the same data loss and recovery time as the 1:1 DPP since traffic is immediately switched over to the backup upon detection and at the same time the receiving and starts to listen on the backup path. With SPP, the PXC reconfiguration (signaling and uploading of maps) costs dominate both data loss and recovery time, bringing data loss to between 50 and 90 ms.

4.3.3 Non-dynamic, Online Provisioning

Our first set of experiments are based on simulations of non-dynamic provisioning with uniform full-mesh traffic demands. We assume that we have no knowledge of future demands. Each connection is assumed to be bidirectional. We used 200 randomly selected orderings of the demands and report the mean value measured over these orderings. The capacity results are shown with 98% confidence intervals in the Figure 4.12 and 4.13, but are omitted from the graph because at 98% confidence level, the mean values vary by less than 1%.

82
To make a fair comparison, the same set of orderings is used in all experiments. Total capacity is the primary metric used. We assign each $\lambda$-channel on each link a cost of one when calculating capacity. The total capacity cost is therefore the sum of the number of $\lambda$-channels reserved for both primary and backup paths. While, in practice, traffic demands may not be uniformly distributed, the study of uniformly distributed demands suffices to illustrate the characteristics of the different protection algorithms for comparison purposes. This method is consistent with other comprehensive online provisioning work found in the literature [19, 17]. Approaches such as design-based routing (DBR) [69] that use offline ILP optimization to provision for online allocation decisions may reduce overall capacity requirements, but we do not expect that they will lead to substantial changes in the relative costs of algorithms. Also, DBR can be applied to many algorithm that shares protection resources including the algorithms evaluated in this work.

**Capacity vs. Average Backup Path Length**

In this section, we report measurements based on shortest primary routing with varying backup path lengths (starting from shortest backup path to backup paths that are arbitrarily longer than the shortest backup path in single hop increments). Path lengths are computed in terms of number of hops. We also present the results using the *average path length expansion ratio* [30] (pl-expansion for short), which provides more insight in terms of the penalty an algorithm pays due to an increase in path lengths. Path length expansion ratio for a connection is the ratio between the allocated backup path over the shortest path length for a given source/destination pair. Average path length expansion ratio, then, is the average over all connections. With shortest (robust) primary paths, pl-expansion is also the average ratio between backup and primary paths for the link failure
protection scenario. For the node failure protection scenario, however, there are few cases where the shortest robust primary is different from the shortest path in the network.

Considering tradeoffs is in terms of path lengths is interesting because it provides an insight as to how much additional delay is introduced. In addition, in the case of AONs, signal qualities as well as possible amplification requirements can be captured. One important issue, often overlooked, with extending paths is that an increase in path lengths (of some number of hops or distance) more greatly affects shorter connections. Pl-expansion captures this effect and provides information about the actual penalty an algorithm may incur in optimizing for capacity. The average path length measure does not show this effect. For example, in Figure 4.10, pl-expansion for Streams is lower than SPP even though the average path length is higher for both NATIONAL and COST 239. This result shows that in SPP connections with shorter paths were assigned to longer paths more often compared to Streams (to allow better use of protection capacity).

**Link Failure Protection**

Figures 4.6 and 4.7 show capacity requirements and average path lengths for different networks and algorithms configured to handle all single link failures (node failure coverage is discussed in the following section). Results corresponding to shortest primary paths, and shortest backup (+0 hop) and arbitrary length backup (N-hop) solutions are shown. Figure 4.6 shows total capacity normalized to that of DPP. Evaluation results show the relative efficiency between different algorithms. SPP and Streams incur a small overhead in path length expansion, with a significant reduction in capacity compared to DPP. Our results confirm that SPP is the most capacity-efficient (well understood in the literature). Streams provides faster recovery and minimal loss with a small capacity overhead compared to SPP.
Figure 4.6: Total capacity normalized to DPP with shortest primary paths for link failure protection.

Figure 4.8 shows total capacity (normalized to best SPP solution) and the pl-expansion. SPP and Streams can be operated with flexibility in terms of these two measures by varying the maximum allowed backup path lengths. We use the number of additional hops compared to the shortest possible backup path to control the tradeoff between capacity and pl-expansion. The full range of solutions from shortest backup (+0 Hop) to arbitrary length backup (N-hop) is shown. The lowest (and right most) data points for each algorithm represent +0 hop solutions with single hop increments to the maximum allowed backup path length shown to the left in series. Naturally, different networks and algorithms have different number of data points as the effect of allowing extra hops for backup paths plateaus at different points. The top most data point corresponds to N-hop solutions (where N may vary for different networks and algorithms). The same
Figure 4.7: Average backup path lengths in hops with shortest primaries. PL-expansion numbers are shown inside the parentheses.

scales are used for all graphs in this figure to aid in visual comparison for the different algorithms and networks. Note that the DPP results for COST-small and NSFNET are off the graph. These figures represent the operating points for the different protection schemes.

For DPP, only shortest backup solutions are presented, as increasing backup path lengths only increases capacity. The pl-expansion value for DPP reflects an absence of disjoint shortest paths for many pairs. For most networks, allowing one or two extra hops in the backup substantially reduces capacity without significantly increasing pl-expansion. This phenomenon is somewhat intuitive given that the possible choices for paths between two nodes increase exponentially with increased maximum allowed length. Our experiments showed that allowing longer primary paths slightly improve each algorithm in terms of capacity due to the increase in the number of candidate primary and backup path pairs, but does not significantly change the relative results between different algorithms.

Node Failure Protection

Figure 4.9 shows the total required capacity (normalized to DPP) and Figure 4.10 shows the path lengths (and pl-expansion) for the different protection algorithms with node protection. SPP and Streams both incur some capacity overhead when
Figure 4.8: Operating points for the different algorithms: Normalized capacity and pl-expansion. The lowest points represent shortest (+0 hop) backup with data points corresponding to additional hops from the shortest.
node failures are considered. The overhead shown by our results is expected as there are fewer number of node-disjoint paths compared to link-disjoint paths, which limits the number of choices for primary and backup path pairs. Figure 4.11 shows the operating points for the different algorithms. The results closer together and capacity gap between SPP and Streams is smaller.

With the added overhead, differences between the algorithms are smaller compared to link failure results. Backup path lengths tend to be shorter compared to the link failure scenario (except for the +0 hop case) because more candidate paths were present in the link failure scenario, and the longer candidates were often chosen to optimize for capacity.

Confidence Intervals

Figures 4.12 and 4.13 show 98% confidence intervals computed using standard statistical methods with 200 random samples used to generate non-dynamic, on-line routing results in the previous sections. As stated previously, the numbers vary by much less than 1%. Note that DPP results are always the same when minimizing capacity-cost as shortest primary and backup path pairs result in the lowest cost pair for each connection.
Figure 4.9: Total capacity normalized to DPP with shortest primary paths for link and node failure protection.

<table>
<thead>
<tr>
<th>Network [node deg.]</th>
<th>Primary</th>
<th>DPP, SPP &amp; Streams (+0 hop)</th>
<th>SPP (N-hop)</th>
<th>Streams (N-hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COST-small [4.36]</td>
<td>1.60</td>
<td>2.27 (1.58)</td>
<td>2.66 (1.89)</td>
<td>2.45 (1.70)</td>
</tr>
<tr>
<td>NJLATA [4.18]</td>
<td>1.75</td>
<td>2.31 (1.49)</td>
<td>2.41 (1.57)</td>
<td>2.36 (1.52)</td>
</tr>
<tr>
<td>COST 239 [3.89]</td>
<td>2.24</td>
<td>3.25 (1.56)</td>
<td>3.61 (1.78)</td>
<td>3.65 (1.76)</td>
</tr>
<tr>
<td>NATIONAL [3.67]</td>
<td>2.92</td>
<td>4.15 (1.58)</td>
<td>4.63 (1.83)</td>
<td>4.72 (1.82)</td>
</tr>
<tr>
<td>ARPINET [3.20]</td>
<td>2.75</td>
<td>4.15 (1.75)</td>
<td>4.64 (2.03)</td>
<td>4.92 (2.13)</td>
</tr>
<tr>
<td>NSFNET [2.86]</td>
<td>2.20</td>
<td>3.73 (2.03)</td>
<td>4.01 (2.22)</td>
<td>4.18 (2.32)</td>
</tr>
</tbody>
</table>

Figure 4.10: Average backup path lengths in hops with shortest primaries. PL-expansion numbers are shown inside the parentheses.
Figure 4.11: Operating points for the different algorithms: Normalized capacity and pl-expansion. The lowest points represent shortest (+0 hop) backup with data points corresponding to additional hops from the shortest.
<table>
<thead>
<tr>
<th>Network [node deg.]</th>
<th>DPP</th>
<th>Streams (0-hop)</th>
<th>SPP (0-hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>N-hop</td>
<td>N-hop</td>
</tr>
<tr>
<td>COST-small [4.36]</td>
<td>426</td>
<td>316.32±0.73</td>
<td>259.71±0.68</td>
</tr>
<tr>
<td></td>
<td></td>
<td>278.32±1.05</td>
<td>239.35±0.55</td>
</tr>
<tr>
<td>NJLATA [4.18]</td>
<td>446</td>
<td>387.34±0.53</td>
<td>351.28±0.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>366.21±0.73</td>
<td>337.57±0.56</td>
</tr>
<tr>
<td>COST 239 [3.89]</td>
<td>1870</td>
<td>1567.57±1.36</td>
<td>1394.76±1.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1452.38±1.91</td>
<td>1329.59±1.17</td>
</tr>
<tr>
<td>NATIONAL [3.67]</td>
<td>3902</td>
<td>3247.75±2.27</td>
<td>2924.28±1.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2963.86±3.30</td>
<td>2794.28±2.02</td>
</tr>
<tr>
<td>ARPANET [3.20]</td>
<td>2614</td>
<td>2101.78±2.13</td>
<td>1842.03±1.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1882.24±2.85</td>
<td>1782.95±1.96</td>
</tr>
<tr>
<td>NSFNET [2.86]</td>
<td>1078</td>
<td>825.89±1.62</td>
<td>684.42±1.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>754.01±2.16</td>
<td>666.69±1.03</td>
</tr>
</tbody>
</table>

Figure 4.12: Total capacity from online provisioning for link failures. 98% confidence intervals are shown for 0-hop and N-hop solutions. DPP cost is always the same since the shortest path pairs yield the lowest cost.

<table>
<thead>
<tr>
<th>Network [node deg.]</th>
<th>DPP</th>
<th>Streams (0-hop)</th>
<th>SPP (0-hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>N-hop</td>
<td>N-hop</td>
</tr>
<tr>
<td>COST-small [4.36]</td>
<td>426</td>
<td>341.52±0.56</td>
<td>333.08±0.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>337.35±0.65</td>
<td>323.59±0.65</td>
</tr>
<tr>
<td>NJLATA [4.18]</td>
<td>446</td>
<td>402.43±0.43</td>
<td>400.33±0.51</td>
</tr>
<tr>
<td></td>
<td></td>
<td>400.23±0.46</td>
<td>396.01±0.56</td>
</tr>
<tr>
<td>COST 239 [3.89]</td>
<td>1874</td>
<td>1675.44±0.95</td>
<td>1598.47±1.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1618.32±1.45</td>
<td>1554.54±1.31</td>
</tr>
<tr>
<td>NATIONAL [3.67]</td>
<td>3902</td>
<td>3533.72±1.55</td>
<td>3322.94±2.77</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3381.50±2.19</td>
<td>3223.67±3.11</td>
</tr>
<tr>
<td>ARPANET [3.20]</td>
<td>2618</td>
<td>2172.51±1.94</td>
<td>2034.10±2.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2054.50±2.65</td>
<td>1993.64±2.27</td>
</tr>
<tr>
<td>NSFNET [2.86]</td>
<td>1078</td>
<td>878.29±1.58</td>
<td>824.20±1.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>841.38±1.75</td>
<td>804.52±1.31</td>
</tr>
</tbody>
</table>

Figure 4.13: Total capacity from online provisioning for link and node failures. 98% confidence intervals are shown for 0-hop and N-hop solutions. DPP cost is always the same since the shortest path pairs yield the lowest cost.
4.3.4 Dynamic Provisioning

Our second set of experiments is based on simulations of dynamic provisioning. We consider a dynamic call-by-call system with Poisson arrivals and exponential call durations. We simulated the protection schemes using 16 wavelengths per link, with 8 wavelengths going in each direction. Again, we assume that we have no knowledge of future demands, and that each connection is bidirectional. The primary metric for the dynamic provisioning simulations is blocking probability, and is measured over varying loads (product of the average arrival rate and the average call duration). We let each simulation run long enough to reach steady state, where the overall number of connections routed is greater than 40000 and up to 240000. The main difference from the non-dynamic provisioning scenario is that wavelength availability varies with time, and shortest primaries may not be available in many cases. We report measurements based on varying primary and backup path lengths by allowing up to two extra hops from the shortest path. Overall, the results are exactly as expected and consistent from studying the non-dynamic model and the relative tradeoffs are also as expected. Details are provided in the following sections.

Link Failure Protection

Figure 4.14 shows blocking probabilities for varying loads for the networks, and Figure 4.15 shows the path lengths when considering link failure protection only. Blocking probabilities are measured to 5% blocking, and path lengths are measured to whatever the maximum load is at 5% blocking for each network using the three protection schemes. In all cases, the SPP algorithm performs best in terms of call blocking, as expected.
The general trend in all of the networks is that Streams sits somewhere in between DPP and SPP in terms of call blocking. Both Streams and SPP pay a small penalty in terms of path lengths. At 5% call blocking, on average, Streams can support about 27% more traffic compared to DPP and SPP can support 50% more compared to DPP.

Some interesting trends are shown in Figure 4.15. The trend in path length as a function of load for the COST-small network differs from that of the other networks. SPP backup lengths sharply increase then slowly drop while the primaries show an upward trend with a slightly sharper change at 30 Erlangs. At first, SPP is able to share resources by selecting longer paths (both primary and backup). Streams also has longer backup paths compared to DPP. Together with the large gap in the blocking probability at the bottom of the curve in the graph shown in Figure 4.14, the results show that both Streams and SPP are able to optimize for load even when the network is sparsely loaded. Longer paths allow the algorithms to better search for shared resources by extending the search space, thereby allowing the algorithm to minimize resource usage.

The backup path lengths for SPP start to drop at around 15 Erlangs. This effect is due to the fact that there are more resources already allocated to protection when connection requests arrive. The abundance of allocated protection capacity equates to more available resources for sharing. The primary lengths for SPP start to increase almost exactly at the point where we start to see some blocking, at 30 Erlangs. This trend is a result of the algorithm choosing longer primary paths to avoid blocking as the number of free wavelengths start to decrease with an increase in load. The sharp rises for Streams primary (at load 40+) and DPP backup (at load 20+) are due to congestion where longer paths are chosen to route around the congested links. The sharp increase occurs at loads where call blocking goes over 5%.
Figure 4.14: Single link failure protection. Blocking probability vs. traffic load (Erlangs).
Figure 4.15: Single link failure protection. Primary and backup path lengths (normalized to the average shortest path length) vs. traffic load (Erlangs).
Path length results for the other networks show less pronounced changes with the degree of change dependent on average node degree. There are more paths in networks with higher average node degree, and algorithms can more dynamically adapt to varying loads as the number of candidate paths increases. The primary path lengths for DPP starts to decrease slightly at 12 Erlangs for ARPANET. This result falls in the range where the load is sufficiently high with higher congestion where shorter paths are less likely to be blocked compared to longer paths. This effect occurs for all networks given sufficiently high loads, but is not visible in the ranges shown for the networks except for the case of DPP on ARPANET.

**Node and Link Failure Protection**

Figures 4.16 and 4.17 show the simulation results for the different algorithms when considering single link and node failures. Similar to the results from the previous section, Streams’ blocking performance fall between DPP and SPP.

The main difference between these results and the results from the previous section is that the two shared protection schemes (Streams and SPP) perform less efficiently as there are fewer path choices. This is due to the fact that there are usually fewer node disjoint paths in a network than link disjoint paths (especially when the network is not sparse).

The graphs are plotted on the same scale as the ones from the previous section to aid in visual comparison of the two scenarios. DPP results are not significantly affected by considering node failures. Considering node failure protection pushes both shared algorithms closer to DPP.

The slope of the changes in path length is strongly related to the slope of the graphs shown in the blocking probability versus load plot. This phenomenon is true for relatively low network loads (at 5% or less blocking in this case). When fewer resources are available, longer paths are considered in an effort to route
Figure 4.16: Single link and node protection. Blocking probability vs. traffic load (Erlangs).
Figure 4.17: Single link and node protection. Primary and backup path lengths (normalized to the average shortest path length) vs. traffic load (Erlangs).
around blocked wavelengths. Once the network becomes more heavily loaded, the algorithms tend to favor shorter paths. This trend is not clear in the results shown because the graphs only show relatively small loads.

4.4 Protecting Mission Critical Traffic With Streams

In [70], classification of traffic based on different survivability requirements for the Department of Defense was specified to allow more flexible and efficient operation of optical networks, and to reduce this cost. Similar work in survivability based service classification that aims at reducing overall cost by differentiating services based on the level of reliability required by different classes of traffic can be found in [71, 59]. However, high-priority, mission critical traffic requires minimal downtime in the event of failures and utilizing dedicated protection to solve this problem can still be expensive and may adversely affect the available budget (and often a compromise is made by utilizing cheaper protection schemes that are slower with more data loss) [70].

Given that Streams offers rapid-recovery and that it can co-exist in the network with other standard protection schemes (such as dedicated or shared path protection), it serves as a good candidate for allowing service differentiated operation of networks similar to the methods outlined in [70, 71, 59] without additional equipment. Dynamic provisioning, as discussed in [71], is attractive because it allows flexible setup and tear down of connections as communication needs vary and is going to play a key role in how future networks are operated. However, for high-priority connections, it may be essential to perform more static allocation of resources to guarantee not only reliable services, but also stable, non-blocking communication. With the goal of minimizing capacity cost while providing fast
recovery (with small data loss), we present an attractive and practical solution using an optimized version of Streams.

In this section, we introduce a novel capacity optimization technique for use in provisioning high-priority traffic. We first introduce an integer linear programming (ILP) based optimization technique to minimize capacity cost. We then present a novel heuristic to reduce the problem space and show that our heuristic is effective in achieving its goal. Simulation results show that protection cost can be reduced by a significant amount (36% or more) using our technique.

### 4.4.1 Capacity Cost Optimization

We now present our optimization technique based on an ILP formulation and also present our stream selection heuristic. For static traffic demands, protection capacity provisioning for Streams can be optimized by solving an ILP problem. In this section, we first present the ILP formulation and address several practical issues in finding optimal solutions. The protection capacity sharing optimization problem is in general difficult to solve as the problem space grows exponentially given the number of choices of backup paths for each connection. For practical applications, limiting this choice to few shortest backup paths while utilizing shortest primary paths has been proposed in the past [10]. Shared protection schemes are less constrained in the way backup paths need to be arranged compared to Streams, and therefore can relatively efficiently utilize the technique mentioned above. However, from our experience in working with Streams in the context of online routing, we noticed that allowing longer backup paths allowed for a greater reduction in capacity cost relative to shared protection. This phenomenon is true also for slower shared protection schemes, but they are affected much less by it due to their flexibility in sharing.
To address this problem, we utilize an approach that is somewhat opposite what has been done in the past—in the context of Streams, instead of merging backup paths into streams, we start with some number of streams and fit backup paths onto them. This approach also can suffer from the same problem as the number of possible streams (paths and cycles in a network) can be huge depending on the network. However, using our Q-stream selection algorithm, we are able to determine the potential efficiency of the streams and limit the problem by utilizing only a small percentage of the total number of the streams. This technique is presented later in the section.

![Figure 4.18: (a) Simple path-stream. (b) Simple cycle-stream. (c) Complex cycle-stream. (d) Complex path-stream.](image)

In a simple graph (i.e., there is at most one edge between any two nodes), there are four types of streams—simple paths and cycles (nodes are repeated at most once for paths and at most once for all nodes except for the start/end node for cycles), and complex paths and cycles (nodes may be visited many times)—as shown in Figure 4.18. Whether a stream is simple or complex is determined by the topology of the subgraph that is induced by the stream. We denote a stream simple if and only if the induced subgraph is either a simple path or a simple cycle. The number of paths and cycles can also quickly grow as the network size and complexity grows, but in practice it is relatively easy to find for backbone network topologies. However, finding all complex streams adds a significant amount of complexity to this problem. In this work, we only utilize simple streams.
4.4.2 ILP Formulation

$E$ set of edges in the network.

$W$ set of available $\lambda$-channels on each edge.

$P$ set of primary paths. $p \in P, p \subseteq E$

$S$ set of streams. $s \in S, s \subseteq E$

$F_p$ feasible set for each primary path $p \in P$. $F_p \subset S$ contains all streams that can be used to protect path $p \in P$.

$F_s$ feasible set for each stream $s \in S$. $F_s = \{p \in P \mid s \in F_p\}$

$CS_e$ set of tuples of conflicting streams. Streams that use edge $e$. $CS_e \equiv \{s \in S \mid e \in s\}$. They cannot be assigned to the same wavelength.

$C$ set of tuples $\{(p_1, p_2, \ldots, p_n), s\}$ of primary paths incompatible on stream $s$. Basically, the same stream $s$ cannot be used to restore all of these primaries.

The elements of $C$ are in $P(P) \times S$ s.t. $\forall c \in C, c = (\pi, s), \pi \subseteq F_s$. By definition, $\{(p), s\}$ is not a member of the set $C$, given $p \in F_s$. $\{(p_1, p_2), s\} \in C$ means that $p_1$ and $p_2$ cannot utilize the same stream $s$, even though they are individually compatible with and can use $s$ s.t. $p_1, p_2 \in F_s$. Similarly, $\{(p_1, p_2, p_3), s\} \in C$ says that the three primary paths cannot share the same stream $s$ even if any two of them can. This definition can be extend to any number of primaries. A detailed discussion on how to find these sets is provided in the next section and is the key to this optimization technique.

$len_s$ length of the stream $s$ (in hops), which is simply the number of $\lambda$-channels required to allocate $s$.

$X_{ps}^w$ indicator for association between a primary path and a stream. It is a binary variable that is set to 1 if stream $s$ allocated on wavelength $w$ is used to protect the primary path $p$ and 0 otherwise.

$Y_{sw}$ indicator for allocation of a stream. It is a binary variable that is set 1 if stream $s$ is allocated on wavelength $w$ and 0 otherwise.
The objective is to minimize the total cost for allocating protection capacity using Streams. Note that we assume that each $\lambda$-channel has a cost of one, but the cost can be easily modified otherwise by replacing $\text{len}_s$ with an appropriate cost function. Ultimately, the solution to the ILP returns a set of $Y_{sw}$’s such that all primaries are covered with minimum total capacity cost.

\[
\text{Minimize } \sum_{s \in S} \sum_{w \in W} \text{len}_s Y_{sw} \quad (4.1)
\]

It is subject to the following constraints. The first constraint enforces that each primary path is assigned to one and only one stream for backup.

\[
\forall p \in P, \sum_{s \in F_p} \sum_{w \in W} X_{wp} = 1 \quad (4.2)
\]

Each stream is considered occupied if at least one primary path has been assigned to it. The binary variable enforces that such assignment is only counted once when multiple primary paths utilize the same stream. Note that the objective function rules out solutions for which $Y_{sw} = 1$ and all $X_{ps}$ are 0 for some $s, w$ pair.

For all $p \in P, s \in S, w \in W$,

\[
Y_{sw} \geq X_{ps} \quad (4.3)
\]

Next, conflicting primaries incompatible with a stream cannot be allocated on the same $\lambda$-channel of that stream. Given a set of conflicting primaries and a stream $s$, $s$ should not be used by all of the primaries in the set. At most $n - 1$ connections can share the same stream, where $n$ is the number of total primaries in the set. Note that it does not imply that any strict subset of the primaries is incompatible with the given stream. A subset of the primaries can share the stream if this subset and the stream are not in the set $\mathcal{C}$. 

103
∀(\{p_1, p_2, \ldots, p_n\}, s) \in \mathbb{C}, w \in W,

\begin{equation}
X_{p_{i}s}^w + X_{p_{i+1}s}^w + \cdots + X_{p_n s}^w \leq n - 1 \tag{4.4}
\end{equation}

Conflicting streams that have any edge(s) in common cannot utilize the same λ-channel for any of their edges. In other words, each wavelength can only be assigned to at most one of the streams in the conflicting set.

∀e \in E, w \in W,

\begin{equation}
\sum_{s \in CS_e} Y_{sw} \leq 1 \tag{4.5}
\end{equation}

Wavelength continuity constraints is implicitly satisfied by the use of the binary variables. Allocation of each stream is done on a per-wavelength basis instead of allocating capacity on a per-edge basis. This constraint forces all edges on the stream to utilize the same wavelength.

### 4.4.3 Finding the Incompatibility Set \(\mathbb{C}\)

In most cases, a single stream can protect multiple primary paths individually (i.e., all primaries in \(F_s\) are protected by \(s\)). However, not all primaries in \(|F_s|\) can be protected simultaneously by \(s\). A set of primaries that can share stream \(s\) simultaneously are said to be compatible (formal definition is provided in the Appendix). Determining the compatibility of a stream for sets of primaries is important for solving the ILP.

Figure 4.19(a) illustrates a simple set of two primaries that are compatible. The two primaries 0-1-4-5 and 2-1-4-3 can both be protected simultaneously by the stream shown in solid gray curve. This stream can protect the two primaries under any single link failure, including the common edge (1,4) since they utilize different parts of the stream. However the primaries shown in Figure 4.19(b) are
Figure 4.19: (a) Two compatible primaries. (b) Incompatible primaries. (c) Incompatible primaries.

Note compatible, as the failure of edge (1,4) requires the two primaries to utilize the same part of the stream. The stream shown in Figure 4.19(c) cannot protect the three primaries (0-4-6, 1-4-6 and 1-4-3) together even though they are pair-wise compatible. In other words, any two combination of the three primaries can be simultaneously protected by the stream, but not all three can be protected.

It is possible to find all incompatible set $C$ for each stream $s$ by checking the compatibility of all subsets of primaries in $F_s$. For each subset, an exhaustive search can be performed recursively. Since a compatible set must have all its subsets compatible, a bottom up approach can be used with memoization to keep track of the states of the subsets. First, for every pair of primary paths in $F_s$, we check the pair-wise compatibility on the stream. If they are not compatible, the set is added to $C$. If at least one two-primary combination passes the check, we can check the compatibility of every three primary combination that includes this two-primary combination that passed the check. The same step is repeated for all combinations of primaries. The number of primary paths a single stream can protect is at most $|E|$, but in practice this number is smaller. However, the search space can still grow rapidly when the set size is large. This naive approach requires $O(2^k)$, $k = |F_s| \leq |E|$ checks in the worst case.

For simple streams we found that, for compatibility checks of a set of primaries $R \subset F_s$, $|R| > 3$, checking all subsets $T \subset R$, $|T| \leq 3$ is sufficient. Therefore, after checking all 2 and 3 combinations of primaries in $F_s$, a set of primaries of
size 4 or greater is compatible with the stream if and only if any subset of size 3 is compatible. This result directly follows Theorem 1 presented in the next section along with a proof. By using this property of simple streams, the complexity can be reduced to polynomial $O(k^3)$.

### 4.4.4 Proof of Simple Streams Property

In this section we show that checking compatibility for sets up to size 3 is sufficient for simple streams (i.e., streams formed from simple paths or cycles). Let $s$ be a simple stream.

**Definition 1 (Primary conflict).** Two primary paths are said to conflict if they have an edge in common, and therefore can fail simultaneously when that edge fails.

**Definition 2 (Backup path).** A backup path ($BP$) of primary path $p$ on stream $s$ is a segment $U \subset s$ of the stream that connects the two end points of $p$. The set of $BP$’s for $p$ is denoted $BP_p$.

**Lemma 1.** Any primary path $p \in F_s$ has exactly one $BP$ on stream $s$, if $s$ is not a cycle.

**Lemma 2.** Any primary $p \in F_s$ has exactly two $BP$ on stream $s$, if $s$ is a cycle.

**Definition 3 (Assignment).** Given a simple stream $s$ and given a multi-set $R$ of primary paths, $R \subseteq F_s$, an assignment for $R$ is a function associating each path $p \in R$ with one of its backup paths on $s$.

**Definition 4 (Active backup path).** An active backup path ($ABP$) for primary path $p$ is the $BP$ chosen for protection use among all paths in $BP_p$: $ABP_p$. 
Definition 5 (Compatible). A multi-set $R$ of primary paths is said to be compatible on $s$ if and only if there exists an assignment of active backup paths for all primary paths in such that for any pair of conflicting primaries, the active backup paths assigned to these two primary paths are disjoint.

Definition 6 (Compatible Assignment). Given a stream $s$ consisting of a simple path (not a cycle), and a multi-set $R$ of primary paths, $R \subseteq F_s$, with $|R| > 2$, $R$ is compatible on $s$ if and only if for any subset $T \subset R$, $|T| = 2$, $T$ is compatible on $s$. Since each path has exactly one BP, assignment is unique and induced assignment has same compatibility.

Theorem 1. Given a stream $s$ consisting of a simple cycle and a multi-set $R$ of primary paths, $R \subseteq F_s$, with $|R| > 3$, $R$ is compatible on $s$ if and only if for any subset $T \subset R$, $|T| = 3$, $T$ is compatible on $s$.

Proof. Assume $R$ is compatible on $s$. Then, there exists a compatible assignment of $R$ on $s$. Let $T \subset R$, $|T| = 3$, and consider the assignment induced on $T$ by such compatible assignment of $R$ on $S$. If the induced assignment is not compatible, there exists two conflicting primary paths $p_1, p_2 \in T$ such that the BP’s assigned to $p_1$ and $p_2$ are not disjoint. But, $T \subset R$, so $p_1, p_2 \in R$ and the original assignment of $R$ on $s$ is also not compatible (Contradiction). Thus, $T$ is compatible on $s$.

Assume $\forall T \subset R$, $|T| = 3$, $T$ is compatible on $s$. Now assume that $R$ is not compatible on $s$. We will construct a set $T$ of three paths such that $T$ is not compatible on $s$. There must exist some path $p_1 \in R$ such that all of $p_1$’s BP’s overlap with some other conflicting path in $R$ for any assignment. $p_1$ has exactly two BP’s on $s$. If every path $p \in R$ has a BP that does not conflict with any other BP for any path in $R$, assigning the non-conflicting BP to each path produces a compatible assignment (contradiction). Thus, there exists some $p_1 \in R$ such that both of its BP’s conflict with BP’s for some other path in $R$.  

107
Let $p_2 \in R$ conflict with $p_1$. There must exist an assignment of backup paths to $p_1, p_2$ so that there is no crossing between their $ABP$’s. Figure 4.20(a) illustrates the case where there is a crossing between the pair, and just the two primaries are not compatible. Therefore, the assignment must be in some general form of the example illustrated in Figure 4.20(b), where only one of the two backup path is viable for a compatible assignment due to the conflict between $p_1$ and $p_2$. Now let $p_3 \in R$ conflict with $p_1$, where $p_3$ may or may not conflict with $p_2$. Again, there must exist an assignment of backup paths to $p_1, p_3$ so that there is no crossing to allow $p_1$ and $p_3$ to be compatible on $s$. Again, the assignment must be as illustrated in Figure 4.20(c) and no viable backup paths exist. Thus, $T = \{p_1, p_2, p_3\}$ is not compatible on $s$.

\[ \square \]

### 4.4.5 Stream Selection and Q-streams Heuristic

The ILP solution time depends strongly on the number of streams considered. Thus we tried to efficiently reduce this number without impacting solution quality. After finding all streams in a network, we first to discard streams that do not provide protection for any primary paths for the given traffic demand. Streams that are left after this step are called *valid streams*. However, given a reasonable
number of connections in the problem set, this step does not eliminate a significant
number of streams.

If the total number of streams is large, we utilize the metric shown in Equation
6 to determine the quality of each stream and use a subset of streams called Q-
streams based on their computed quality (Q).

\[
Q_s = \frac{1}{len_s} \left( \sum_{i=1}^{\left| F_s \right|} i^2 v_{s,i} \right)
\]  

(4.6)

where \( v_{s,i} \) is the number of compatible sets of size \( i \) in \( F_s \). Our metric basically
measures how many combinations of primaries in \( F_s \) a stream can protect with
more weight given to compatible sets with higher number of primaries.

![Networks](image)

**Figure 4.21: Networks**

<table>
<thead>
<tr>
<th># simple streams</th>
<th>NSFNET</th>
<th>vBNS</th>
<th>COST</th>
<th>NJ LATA</th>
<th>ARPA NET</th>
</tr>
</thead>
<tbody>
<tr>
<td>vBNS</td>
<td>1209</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSFNET</td>
<td>7370</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NJ LATA</td>
<td>9510</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARPA NET</td>
<td>198013</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COST</td>
<td>552105</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Q-stream efficiency](image)

**Figure 4.22: Q-stream efficiency**
Three of the five networks shown in Figure 4.21 were small enough to allow the ILP to complete with all simple streams as input in a short amount of time (less than about an hour using the CPLEX software on HP Blade servers with AMD Opteron 2000 series processors and 2GB of RAM). The total number of simple streams are shown on the left in Figure 4.22 for all five networks. The right side of the figure shows the efficiency of our Q-stream selection heuristic. The horizontal axis represents the fraction of the total number of simple streams with the highest Q values used as input to our ILP, and the vertical axis represents protection capacity cost normalized to the optimal case where all simple streams were used. We also added the shortest backup paths to the set of streams used for the ILP when the shortest backup paths are not already included by using the Q-stream selection. This method allows the ILP to find more efficient solutions using a small fraction as it has the freedom to leave out some connections while attempting to protect as much primaries as possible with longer streams. The connectivity of NJLATA network is much higher compared to the vBNS and NSFNET networks, and the performance of Q-streams is better for NJLATA. For all cases, however, it shows that the Q-streams heuristic is efficient in reducing the problem size while allowing the solver to find very good solutions.

4.4.6 Summary of Results

We used five well-known networks, shown in Figure 4.21, that are representative of optical backbone network topologies. 20 different provisioning scenarios for each network consisting of 30 randomly selected bi-directional connections for each scenario were performed. We used shortest robust primary paths—when topological shortest paths do not have a disjoint path, the shortest of paths that have disjoint paths must be used. vBNS required the use of shortest robust primaries that are
Table 4.1: Simulation Results.

<table>
<thead>
<tr>
<th></th>
<th>vBNS</th>
<th>NSFNET</th>
<th>ARPANET</th>
<th>COST</th>
<th>NJLATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>node deg.</td>
<td>2.83</td>
<td>2.86</td>
<td>3.20</td>
<td>3.89</td>
<td>4.18</td>
</tr>
<tr>
<td># streams</td>
<td>1209</td>
<td>7370</td>
<td>198013</td>
<td>552105</td>
<td>9510</td>
</tr>
<tr>
<td># valid streams</td>
<td>993</td>
<td>6169.1</td>
<td>136172.1</td>
<td>376984.2</td>
<td>8740.0</td>
</tr>
<tr>
<td>±35.9</td>
<td>±270.2</td>
<td>±7614.1</td>
<td>±24301.4</td>
<td>±141.2</td>
<td></td>
</tr>
<tr>
<td># streams used for ILP</td>
<td>993</td>
<td>6169.1</td>
<td>6837.7</td>
<td>7568.6</td>
<td>8740.0</td>
</tr>
<tr>
<td>±35.9</td>
<td>±270.2</td>
<td>±380.6</td>
<td>±485.9</td>
<td>±141.2</td>
<td></td>
</tr>
</tbody>
</table>

Capacity Cost

<table>
<thead>
<tr>
<th></th>
<th>Primary only</th>
<th>1:1 dedicated protection</th>
<th>Streams Greedy +0</th>
<th>Streams Greedy +N</th>
<th>Streams Optimized</th>
<th>Improvement over Greedy</th>
<th>Improvement over 1:1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>149.8 ± 5.3</td>
<td>240.6 ± 8.1</td>
<td>201.2 ± 9.7</td>
<td>188.8 ± 12.7</td>
<td>153.0 ± 9.34</td>
<td>18.9%</td>
<td>36.4%</td>
</tr>
<tr>
<td></td>
<td>128.4 ± 3.0</td>
<td>221.8 ± 3.5</td>
<td>159.6 ± 5.6</td>
<td>140.6 ± 4.3</td>
<td>81.8 ± 4.4</td>
<td>41.8%</td>
<td>63.1%</td>
</tr>
<tr>
<td></td>
<td>170.2 ± 6.4</td>
<td>250.6 ± 6.9</td>
<td>200.0 ± 5.1</td>
<td>165.6 ± 6.7</td>
<td>143.2 ± 4.4</td>
<td>13.5%</td>
<td>42.8%</td>
</tr>
<tr>
<td></td>
<td>135.4 ± 3.6</td>
<td>194.2 ± 4.1</td>
<td>161.0 ± 5.8</td>
<td>145.8 ± 6.3</td>
<td>124.4 ± 7.6</td>
<td>14.7%</td>
<td>35.9%</td>
</tr>
<tr>
<td></td>
<td>106.8 ± 2.5</td>
<td>139.4 ± 2.7</td>
<td>112.8 ± 3.7</td>
<td>102.8 ± 3.4</td>
<td>76.8 ± 3.4</td>
<td>25.3%</td>
<td>44.9%</td>
</tr>
</tbody>
</table>

longer than topological shortest paths. Our results are summarized in Table 4.1 along with 95% confidence intervals. The top portion of the table shows the average node degree, total number of simple streams, total number of valid streams, and the total number of streams actually used in the ILP. For the ARPANET and COST network, we utilize the Q-stream heuristic to reduce the number of streams using only 5% and 2% of the total number of valid streams respectively. The lower portion of the results table first show the capacity cost for provisioning primary paths only, followed by cost for using dedicated path protection, online greedy Streams with shortest backup path only and arbitrarily long backup paths (shown for reference only using algorithm presented in [64]). The results for greedy
Streams are obtained by taking 30 random demands in routing them in random orderings, provisioning streams that greedily minimize cost. The results are averaged over 50 random orderings for each of the 30 random demands. Finally, capacity cost for using ILP optimized Streams is shown along with the amount of improvement over the greedy approach as well as the improvement over dedicated protection. The results for ARPANET and COST networks are obviously not optimal (given utilization of only 5% and 2% of the available streams), but our heuristic is efficient in that it allows the ILP solver to find good solutions that are comparable to improvements shown in other networks. Overall, it shows that a significant savings in capacity cost can be achieved using the technique presented in this thesis.

4.5 Summary

Most survivability algorithms developed for optically opaque networks are not readily applicable to all-optical networks and result in lower efficiency due to added limitations such as the \( \lambda \)-continuity constraint. The Streams algorithm is attractive because it can be readily implemented in existing networks and can co-exist with algorithms such as SPP and DPP without additional hardware support and with a small management overhead.

Our results show that the Streams algorithm is efficient in terms of capacity usage, allowing us to meet the goal of designing low loss survivability techniques that are capacity-efficient. The Streams algorithm’s recovery speed and data loss characteristics are identical to 1:1 DPP, but is significantly more efficient in utilizing available wavelengths. Therefore, it offers an attractive tradeoff between recovery speed, data loss and capacity utilization. Furthermore, we also showed that Streams can be applied to offline provisioning scenarios with known, static
traffic demands using ILP solutions as well as heuristics to achieve higher efficiency in capacity usage.

The bottom line is that Streams can be used instead of 1:1 protection to achieve the same level of recovery speed while significantly reducing capacity cost or blocking probability.

There are couple interesting questions, though outside the scope of this work, that stem from our work. To evaluate and compare different protection schemes, in addition to the fundamental tradeoffs covered in this work, it may be interesting to experiment with how well these algorithms perform under different types of failure models, such as multiple failures. It may also be interesting to study the impact of online re-optimization techniques for backup capacity, but it may significantly increase management overhead for any protection scheme.
CHAPTER 5

Impact of Physical Layer Impairments on Reliability

As discussed in motivating the need for a high-speed recovery protocol such as the Streams algorithm presented in the previous chapter, all-optical networks (AON) offer several critical advantages over optically opaque networks. Higher data rates can be achieved with the absence of electronic and photonic processing delays that can act as a bottleneck on the total transmission time. Unlike networks that utilize optical-electronic-optical (OEO) conversion, AONs can handle signals with different data rates, protocols, and formats, allowing flexibility in architectures that are capable of supporting changes in today’s standards [23, 63]. However, they suffer from non-ideal physical transmission and component characteristics, which bounds the network performance. A number of transmission impairments that AONs are susceptible to include signal degradation, amplified spontaneous emission (ASE) noise, gain saturation and crosstalk. A few of the factors contributing to signal degradation in AONs include fiber attenuation losses, switch and mux/demux insertion losses and tap losses. In order to compensate for the signal degradation caused by these losses, amplifiers are utilized to boost the signal strength to a level where it can be successfully detected by photodetectors within the network. The use of erbium-doped fiber amplifiers (EDFA) to boost signal levels, however, also causes the generation of ASE noise that accumulates along a lightpath with every amplification. Another impairment brought on by the use of EDFAs is that of gain saturation. As the strength of the signal increases the added gain of the EDFA saturates to a lower amount lending to further sig-
nal degradation as the signal traverses the network. In traversing the network, multiple signals cross paths in a switching node where they can interfere with each other and generate crosstalk. All of the above mentioned transmission impairments adversely affect the bit error rate (BER) of connections and become a bottleneck in network performance [72].

Various performance studies along with novel routing schemes under physical impairments in AONs can be found in the literature [73, 74, 75, 76]. To avoid significant drops in signal to noise ratios (SNR), connections are spaced out across the network to better optimize against crosstalk and improve blocking performance. A performance study using BER estimation for all candidate connections is presented in [73]. To reduce the computation time during online, dynamic provisioning, a crosstalk component counting technique is proposed in [74]. Using the outcome of this weighted counting technique, BER estimation is performed once per connection request, significantly reducing provisioning overhead. In [75], a simple and efficient wavelength ordering technique is proposed to improve QoT unaware routing blocking probability.

Transmission impairment issues, coupled with the fact that the impact of failures in AONs is exacerbated by their high traffic volumes, necessitate an understanding of how survivability algorithms are affected by non-ideal QoT parameters. With the extremely high volume of traffic carried on wavelength division multiplexed (WDM) networks, failures such as fiber cuts can result in a loss of huge amounts of data and revenue. In order to maintain high quality services for future communication needs, we must be able to guarantee a desired level of robustness. The problems with utilizing protection schemes under physical layer impairments are two-fold. With traditional, QoT unaware protection schemes, backup paths may not meet the BER requirements when they are activated. In addition, activating backup paths can increase the BER on other connections
that have not been directly affected by the link failure. To quantify these phenomena, we introduce two metrics namely QoT vulnerability and Cascading Failure Probability. In [77], optimization and heuristic techniques for maximizing resource sharing for shared path protection for non-dynamic network with sparse OEO conversion was presented. In [78], dynamic AON performance under two dedicated path protection schemes (as later described in Chapter 1) was studied (in terms of both blocking and the probability of connections not meeting the QoT requirements after a failure).

We present different QoT aware schemes based on dedicated path protection (SPP) and shared path protection (SPP). We then compare the various methods in terms of call blocking, QoT vulnerability and cascading failure probability. Our results show that, in general, blocking performance of QoT aware provisioning with protection is close to QoT aware unprotected provisioning scenarios. Under ideal conditions, blocking performance of routing protected connections can be anywhere from 50% to over 100% worse than routing unprotected connections. However, to guarantee 100% survivability, a more constrained routing and wavelength assignment (RWA) technique must be used and our results show that a considerable more calls must be rejected.

The rest of this chapter is organized as follows: First, the underlying network architecture, various sources of transmission impairments, and how they all affects survivability is presented next. Second, the metrics used for performance evaluation along with the techniques used for protection routing are presented in Section 5.2. Section 5.3 summarizes the physical layer models used in this work. Simulation results are provided in Section 5.4 followed by a summary and discussion on future directions in Section 5.5.
5.1 Network Architecture, Crosstalk, and Survivability in AONs

Figure 5.1: AON Node architecture.

In this section, we discuss the underlying network architecture and various sources of impairments that adversely affect the overall QoT. We then discuss how it affects end-to-end protection schemes and network survivability.

5.1.1 Network Model

The components that make up an AON are depicted in Figure 5.1. As shown in the figure, lightpaths traversing through the network propagate through fiber cables and nodes en route to their destination. The lightpaths originate from lasers within the node through the modulation of signals from the electrical domain into the optical domain and are sent to a destination where a photodetector converts the received optical signal back into the electrical domain. As the signal travels through fiber links on to its destination, it is subject to degradation due to fiber attenuation loss brought on by imperfections, absorption and scattering effects.
Figure 5.2: Strictly non-blocking 8x8 switch architecture built using 2x2 couplers. The table shows the number of copies of NxN Banyan required to build a strictly non-blocking switch. Three 8x8 Banyan crossconnect design is used to meet the requirements of a strictly non-blocking switch. By design, the combiners and splitters before and after the core Banyans are disjoint among lightpaths. Therefore, crosstalk (1st order) occurs only in the core.

within the fiber. At each node along the route, the signal passes through an optical fiber tap and experiences an insertion loss. To make up for the combined tap and fiber losses, an EDFA input gain is used to boost the signal strength before the signal enters the switch where the signal is again subjected to additional losses. The losses within the switch arise from demultiplexer and multiplexer insertion losses as well as switch-size dependent insertion and coupling losses from the switch. Upon exiting the switch, an additional EDFA output gain is applied to the signal to help remedy these losses before the signal is sent through an optical fiber tap and onto the next node in the path.
Figure 5.2 illustrates the Banyan network used inside 8x8 switch for each wavelength. In order for the Banyan architecture to be strictly non-blocking (a path exists through the switch from all open in-ports to all open out-ports), multiple planes or copies must be utilized. In [79], the exact number of planes required for various switch sizes is presented. The table in the figure shows the number of planes required to build a strictly non-blocking switch and a crosstalk-free switch (no two paths share a 2x2 coupler) for the node sizes found in the network topologies used in this work.

5.1.2 Crosstalk Model

The two types of crosstalk that can occur in optical crossconnects are in-band crosstalk and out-band crosstalk. The latter type of crosstalk occurs between signals of different wavelengths and outside the passband of an optical filter, thereby allowing it’s harmful effects to be mitigated through the employment of narrow-band optical filters. In-band crosstalk, however, arises in switches from lightwaves using similar wavelengths and can be either coherent or incoherent. Coherent crosstalk occurs when the interfering signals are phase-correlated and incoherent crosstalk occurs when the signals or not phase-correlated. Due to the uncorrelated phase, incoherent crosstalk can have deleterious effects on affected signals.

Figure 5.3: Demux/mux crosstalk.
As the effects of out-band crosstalk can be reduced to acceptable levels, only in-band crosstalk is considered in the work presented here. The two kinds of in-band crosstalk that are considered include switch and demux/mux crosstalk. To reflect the different impacts of the kinds of crosstalk, weights are applied to each kind of crosstalk based upon the switch channel isolation and port isolation characteristics of the switching node. Switch crosstalk occurs as a result of power leakage between the ports of switching modules while demux/mux crosstalk occurs as a result of insufficient adjacent-channel isolation. The former kind of crosstalk occurs when a number of lightpaths, using the same wavelength, impart a portion of their signal to each other as they pass through the switch. Demux/mux crosstalk can occur in two different ways. First, it can occur when lightpaths using different wavelengths ingress and egress using the same ports. As figure 5.3 depicts, when lightpaths LP1 and LP2 pass through the same demultiplexer they interfere and impart a portion of their signal onto each other only to have the leaked signal portion imparted back onto themselves in the form of crosstalk as the signals traverse through the same multiplexer. With this type of crosstalk, the resulting crosstalk might be coherent because the original signal is interfering with itself. However, as the signal traverses through the node, varying propagation delays can bring about effects similar to those of incoherent crosstalk [80]. As with [74] we simplify our model by considering all instances of this type of crosstalk to be incoherent as they make up a small portion of all crosstalk.

Another mechanism that generates demux/mux crosstalk follows a similar mechanism and differs in one aspect, i.e. the signal generating the crosstalk. In this circumstance, signals utilizing different wavelengths interfere with each other while passing through demultiplexers. On passing through multiplexers in the node, they will generate incoherent in-band crosstalk when interfering with signals that use the same frequency as the signal they previously interfered with.
5.1.3 Amplifier Spontaneous Emission Noise

As mentioned earlier, optical signals propagating through the network are subject to losses from a variety of network components. In order to compensate for these losses, amplifiers are placed within the network to boost the signal strength. A side effect of optically amplifying signals, however, is the accumulation of ASE noise which is generated with every amplification thereby posing a limit to the benefits of performing signal amplification. The expression for ASE noise power [72] is given by

\[ P_n = 2n_sp h \nu (G - 1) B_o \]  

(5.1)

In the expression above \( n_sp \) is the spontaneous emission factor, \( h \) is Planck’s constant, \( \nu \) is the optical frequency (where \( \nu = c/\lambda \) and \( c \) is the speed of light) and \( B_o \) is the optical filter bandwidth. The generation and accumulation of ASE noise within propagating signals therefore makes an additional consideration when determining the BER of a lightpath.

5.1.4 Gain Saturation

Another source of transmission impairment from EDFA use comes in the form of gain saturation where the output gain of the amplifier varies depending upon the level of the input optical signal. Lower signal power levels input into the amplifier induce a gain level close to the small signal gain of the amplifier while higher levels of input induce saturated gain levels. Due to the saturated gain of the EDFA the optical signal does not fully recover from the losses it is subject to while traversing the network.
5.1.5 QoT and Survivability

Given a realistic network model with physical layer impairments such as ASE noise and various types of crosstalk, protection schemes can degrade and may not achieve 100% failure recovery. When a network link fails, the backup paths of the connections affected by the failure become active and start to interfere with existing connections. As the newly activated connections induce additional crosstalk, several problems arise. The example shown in Figure 5.4 illustrates how a new source of crosstalk (that did not exist during normal operation) is introduced. In this example, there are two primary paths each with a backup path—$P_1 - B_1$ and $P_2 - B_2$. They are allocated on $\lambda_1$. During normal operation, $B_1$ and $B_2$ are not active and have no impact on transmission. After link $(0,1)$ fails, $P_1$ is turned off and $B_1$ becomes active. In the switch at node 2, crosstalk is introduced between $P_2$ and $B_1$ as both lightpaths cross the same coupler at the last stage of the Banyan. Crosstalk induced by switching from primary to backup paths can generally lead to two problems. First, the backup path’s BER can be too high resulting in an unsuccessful recovery (we term this effect QoT Vulnerability). Second, existing primary paths that are unaffected by the link failure can also fail due to newly introduced crosstalk (we term this Cascaded Failure). Two metrics that quantify these effects are introduced and discussed in Section 5.2.1.

Protection paths are generally also more sensitive to transmission impairments. Depending on the network topology, selecting alternate paths for protection requires the use of longer paths (in terms of hops). First, longer paths can be a problem as more amplification is required inducing additional ASE noise coupled with effects of EDFA saturation. Longer paths are also more susceptible to crosstalk as it crosses more nodes and is more likely to experience crosstalk leading to higher QoT Vulnerability as well as Cascaded Failures.
Figure 5.4: Crosstalk is introduced in the switch between primary path $P_2$ and the backup path $B_1$ when $B_1$ is turned on after the link failure.

5.2 QoT and Protection

Dedicated path protection (DPP) and shared path protection (SPP) are two of the most commonly used end-to-end survivability schemes. There are many other protection schemes that offer various tradeoffs suited to fill different needs of the network. However, DPP and SPP represent the two ends of the spectrum in the critical tradeoff between recovery speed and capacity usage. DPP offers SONET ring-like recovery, while SPP has been shown to significantly reduce operation cost in terms of capacity. In dynamic networks this reduction in capacity equates to the ability to support higher loads at the same level of call blocking. We use DPP and SPP schemes to study the impact of physical layer impairments on survivable optical networks. First we introduce two metrics that allows us to capture the post-failure impact of non-ideal physical layer characteristics that are not present.
in OEO networks. We then present various approaches to performing protection routing under with QoT consideration.

5.2.1 Metrics

QoT-Vulnerability ($V_{QoT}$) measures the likelihood of experiencing incomplete failure recovery due to transmission impairments (despite the fact that a protection schemes is used to provide 100% recovery from all single link failures). Given that we are studying protection schemes (where the network is designed to survive all single link failures), it is more interesting to see how often the network cannot achieve complete recovery rather than measuring the recovery ratio (more useful for measuring restoration techniques where capacity is limited or possibly multiple failure scenarios). It is defined as

$$V_{QoT} = \frac{1}{|E|} \sum_i f_b(i)$$

(5.2)

where $f_b(i)$ is defined as

$$f_b(i) = \begin{cases} 
0 & \text{all live backup paths meet BER requirements} \\
1 & \text{after failure of link } i. \\
1 & \text{otherwise}
\end{cases}$$

(5.3)

Cascaded Failure Probability is defined as the probability of connections (that are not directly affected by a network failure) failing because of the increase in BER due to crosstalk induced by activating backup paths of other connections. It can be computed by using the following formula, similar to $V_{QoT}$.

$$V_{cas} = \frac{1}{|E|} \sum_i f_p(i)$$

(5.4)
where $f_b(i)$ is defined as

$$f_p(i) = \begin{cases} 
0 & \text{all live primary paths meet BER requirements after failure and recovery of link } i. \\
1 & \text{otherwise}
\end{cases}$$

(5.5)

In this work, we assume equal likelihood of link failures, but the metric can easily be adapted given specific failure rates for each link by simply factoring in coefficients for each link. Note that the change in failure rates does not affect the contribution of this work. Further studies incorporating specific failure rates for specific networks may be interesting, but is outside the scope of this work.

Another metric commonly used to study survivability is failure ratio (or recovery ratio = 1 - failure ratio). This metric is used to study restoration schemes or failure scenarios not covered by the protection model to quantify the impact of such failures (for example, multiple failures). Given that we are dealing with protection schemes, it is much more critical to understand whether or not, and how likely a network is unable to perform as designed (100% survivability, especially given that a significant cost overhead is incurred to build protection schemes into the network) rather than studying how much of the network can be restored. However, we use the failure ratio metric to understand how well a network can do without utilizing a QoT-guaranteed protection scheme. Failure ratio is defined as the ratio between the number of connections that are not recovered due to high BER to the number of connections that are affected by a link failure. It is averaged over all single link failures.
5.2.2 Algorithms

Figure 5.5 illustrates the online call admission process. Upon arrival of a connection request, RWA is performed according to the various QoT-based methods described later in this section. If the chosen path is not available on the selected wavelength, the call is blocked. Otherwise, we enumerate through each lightpath in the network to estimate the BER. If the BER of any lightpath does not meet the minimum requirement, the call is rejected. In non QoT-guaranteed schemes, all primary lightpaths are tested. The method used for the QoT-Guaranteed scheme is described later in this section. If the BER requirement is met for all tested lightpaths, the call is admitted.
In selecting paths, both DPP and SPP require the use of link-disjoint paths for the primary and backup paths. We utilize shortest topological primaries and shortest backup paths (disjoint from the primary). This is a very well understood topic and more details can be found in [10]. Given multiple path and wavelength choices, we perform RWA according to the techniques described below.

- **QoT Aware, Unprotected (Unp)** - We perform RWA for unprotected connections to show how much blocking performance changes when survivability is introduced. We use a crosstalk counting technique to select the path that is more likely to pass the BER requirements. Each type of crosstalk components are counted on each candidate path and a candidate wavelength and the path-wavelength pair with the minimum weight is chosen. If none of the wavelengths are available, the connection is blocked. Note that each crosstalk component is weighted according to the port and switch isolation parameters using the technique introduced in [74]. Candidate selection is done randomly among equally weighted choices. Once the best candidate is found, BER estimation is performed on all existing connections (in addition to the new connection) that are affected by the introduction of the new connection to see if they all meet the BER requirements. The call is rejected if any of the tested connections fail the BER test.

- **QoT Unaware (QoT-UA)** - RWA is performed for each DPP and SPP without consideration for QoT. The goal for SPP here is to minimize capacity usage by maximizing sharing. We greedily select the minimum cost path and wavelength using a method consistent with previous work found in the literature [19]. If no wavelengths are available, the call is blocked. Otherwise, BER estimation is done on the primary path only (similar to unprotected case described above).
• **QoT Aware (QoT-A)** - RWA is performed similar to the technique used for *Unp* for the primary paths. Once the best primary is selected, backup paths are chosen differently for DPP and SPP. For DPP, the corresponding backup path is selected using random placement among available paths and wavelengths. For SPP, backup paths are chosen to minimize the use of new wavelength allocation similar to *QoT-UA*. If there are more than one minimum “cost” backup paths, it is randomly chosen. The final call admission using the BER estimation process is the same as in *QoT-UA*.

In addition, at each node in each path-wavelength combination, we select the copy of the Banyan plane (refer to Figure 5.2) that has the minimum number of crosstalk. Depending on the switch size, several planes exist allowing us to choose the best one. Again, if multiple, equal “cost” options exit, we choose randomly. For both DPP and SPP, note that the Banyan plane selection is done upfront during the RWA stage for both the primary and the backup path. Note that recovery protocol for SPP requires signaling of the intermediate nodes along the path to configure the switch before sending out the data over the backup path. Taking advantage of this process, we can select the best Banyan plane if and when there is a failure, as the intermediate nodes are reconfigured in an attempt to reduce crosstalk.

• **QoT Guaranteed (QoT-G)** - Finally, in order to guarantee 100% survivability, we must consider all single-link failure cases. BER requirements must be met by all connections, under all single-link failures. As with QoT aware primary path selection, BER estimation on all possible backup paths are avoided by using a method similar to crosstalk component counting used above. We extend this approach for backup paths as follows. For each backup candidate, all link failure scenarios are enumerated and the number
Figure 5.6: Crosstalk counting for potential backup path selection for SPP. All possible failure cases must be enumerated. One case shown: link (1,2).

If link (1,2) fails:

- de/mux crosstalk
- switch crosstalk

Possible RWA of P3 & B3

<table>
<thead>
<tr>
<th>connection</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1 @node 3</td>
<td>Xsw</td>
</tr>
<tr>
<td>B2 @node 5</td>
<td>Xdm</td>
</tr>
<tr>
<td>B2 @node 4</td>
<td>Xdm</td>
</tr>
</tbody>
</table>

total count for this case = 3

crosstalk component counting for B3 on λ₁ for (1,2) failure

of crosstalk possible in each scenario is counted. We then select the candidate with the minimum number. Figure 5.6 illustrates this method. Two connections P1 and P2 are already on the network with the corresponding backup paths. We are trying to route P3 and B3. The candidate backup, B3, is on λ₁. Given that P1 and P3 do not fail at the same time, the wavelength on link (5,4) can be shared with B1 for SPP. The bottom part of the figure shows where crosstalk can occur when the potential failure of link (1,2) is considered. P1, B2 and B3 are active, and the various crosstalk sources can be counted. Once the best candidate is chosen, we move on to the BER estimation step.

A path is represented as a set of links in the graph

$E$ - set of all links in the graph

$R(i)$ - set of all connections on the network, where $p \in R(i), \ i \in p$

$R'(i)$ - set of all connections on the network, where $p \in R'(i), \ i \notin p$

$BER(i, \lambda)$ - function returns the estimated BER for path $i$ on $\lambda$
Check backup path \( b \) on \( \lambda \):

1: \textbf{for all} \( e \in E \) \textbf{do}

2: \hspace{1em} \textbf{for all} \( i \in R(e) \) \textbf{do}

3: \hspace{2em} \text{remove} \( i\.primary \)

4: \hspace{2em} \textbf{if} \( BER(b, \lambda) < \text{threshold} \) \textbf{then}

5: \hspace{3em} \text{reject call and stop}

6: \hspace{1em} \textbf{for all} \( i \in R(e) \) \textbf{do}

7: \hspace{2em} \textbf{if} \( BER(i\.backup, i\.\lambda_b) < \text{threshold} \) \textbf{then}

8: \hspace{3em} \text{reject call and stop}

9: \hspace{1em} \textbf{for all} \( i \in R'(e) \) \textbf{do}

10: \hspace{2em} \textbf{if} \( BER(i\.primary, i\.\lambda_b) < \text{threshold} \) \textbf{then}

11: \hspace{3em} \text{reject call and stop}

12: \text{accept call}

For each failed link, we can divide the network into two types of connections—failed connections and connections unaffected by the link failure. For each failed connection, we add the backup paths to the BER estimation engine. For the unaffected connection, we add the primary paths to the same engine. BER estimation is then performed on all connections. If any connection does not meet the BER requirements, the call is rejected. The above algorithm illustrates how this step is enumerated given a chosen backup path and an available wavelength. Overall BER estimation is performed \(|E|\) times more compared to \( QoT-A \).
5.3 Physical Layer Model

5.3.1 Signal, Crosstalk and Noise Power

From earlier discussions, it is apparent that there are three components to the propagating signal that need to be addressed when considering the BER, i.e. the desired signal, crosstalk and ASE noise. The effects of the three components are also shaped by the amplifier gains and various losses the signal goes through while traversing the network. The accumulated gains and losses, $A_{GL}$, that the optical signal witnesses enroute to node $i$ can be expressed as such

$$A_{GL}(i) = L_f(i - 1,i)L_{tap}G_{in}(i)L_{dm}(i)L_{sw}(i) \cdot L_{mx}(i)G_{out}(i)L_{tap}$$

where the $L_x$ and $G_x$ terms in the above equation represent, respectively, losses and gains from the taps, fiber, demultiplexer, multiplexer and EDFAs. Using this equation, the powers of the propagating signal components $P_{sig}(i)$, crosstalk $P_{xt}(i)$ and ASE noise $P_{ase}(i)$ at the output of the $i$th intermediate node can be realized by the following equations:

$$P_{sig}(i) = P_{sig}(i - 1)A_{GL}(i)$$

$$P_{xt}(i) = P_{xt}(i - 1)A_{GL}(i) + \sum_{j=1}^{J_i} X_{sw} P_{in}(j, i)L_{sw}(i)L_{mx}(i) + \sum_{k=1}^{K_i} X_{md} P_{in}(k, i)L_{mx}(i)$$

$$P_{ase}(i) = P_{ase}(i - 1)A_{GL}(i) + 2n_{sp}[G_{in}(i) - 1]h$$

$$\cdot \nu_t B_o L_{dm}(i)L_{sw}(i)L_{mx}(i)G_{out}(i)L_{tap} + 2n_{sp}[G_{out}(i) - 1]h \nu_t B_o L_{tap}$$
The first term of equation (5.8) accounts for crosstalk being accumulated as the signal traverses the lightpath. The second and third terms on the other hand describe the crosstalk generated within the node where \( P_{in}(j, i) \) and \( P_{in}(k, i) \) represent the powers of each of the \( J_i \) and \( K_i \) signals that are sources of switch and demux/mux crosstalk, respectively, within the switch. \( X_{sw} \) represents the switch crosstalk ratio while \( X_{md} \) signifies the mux/demux crosstalk ratio. In the expression for \( P_{ase}(i) \), the first term also accounts for the accumulated ASE noise throughout the lightpath while the second term describes the ASE noise contribution of the in-gain, \( G_{in} \), and the third term does the same for the out-gain, \( G_{out} \), contribution within the current node.

**EDFA Gain Model**

In order to provide realism to the amplifiers used in the node models, the use of ideal amplifiers was abandoned in favor of using amplifiers susceptible to gain saturation and noise generation. The expression for the ASE noise generated by the amplifier is given by equation (5.1) where gain values for the in-gain, \( G_{in} \), and the out-gain, \( G_{out} \), amplifiers are based on a black box EDFA model from [81]. The gain model used in this work is expressed as

\[
G = \frac{G_o}{1 + \left( \frac{P_{in}}{P_{sat}} \right)^\alpha} \quad (5.10)
\]

where \( G_o \) is the small-signal gain, \( P_{in} \) is the input signal power, \( P_{sat} \) is the output saturation power and \( \alpha \) is a parameter used for adjusting the gain saturation. The small signal gain values for the input amplifiers of each node were chosen to offset switch losses, \( L_{sw} \), for the 4x4, 8x8 and 16x16 switches used in the simulation while small signal gain for output amplifiers was kept constant at 22 dB to offset fiber link (\( L_f \)) and tap (\( L_{tap} \)) losses. Values for \( P_{sat} \), the output power where the
gain falls 3 dB, and $\alpha$ where chosen to mimic features of the EDFA gain model used by the authors of [73].

### 5.3.2 Bit Error Rate

The signal, crosstalk and ASE noise powers received at the photodetectors in the network are used to calculate the Bit Error Rate of the lightpath at the respective node and to determine if the lightpath can successfully deliver a signal. The calculated BER at the photodetector is compared to a threshold and call requests that fail to meet the threshold are subsequently blocked. BER calculations in our simulation are done on a per call basis and carried out only if a call has not been blocked due to wavelength availability. The received powers within each node come from lightwaves impinging upon the photodetectors of the node. A simple expression for a lightwave used by the authors of [73] is given by

$$E_R(t) = A \cos(2\pi \nu_i t + \theta(t)) + E_{xt}(t) + E_{ase}(t)$$  \hspace{1cm} (5.11)

The $E_{xt}(t)$ and $E_{ase}(t)$ terms of the expression represent the contributions of crosstalk and ASE, respectively. The first term represents the signal contribution of the lightwave where $\nu_i$ is the frequency, $A$ is the amplitude and $\theta(t)$ denotes the phase of the signal. The photocurrent produced by the impinging lightwave, $E_R(t)$, can be expressed as:

$$i_p(t) = R_\lambda \langle E_R^2(t) \rangle + i_{sh}(t) + i_{th}(t)$$ \hspace{1cm} (5.12)

In the expression above (given by [73]), the shot noise contribution from the lightwave is represented by $i_{sh}(t)$ and the thermal noise contribution of the photodetector is denoted by $i_{th}(t)$. The responsivity of the photodetector is $R_\lambda$, and $R_\lambda \langle E_R^2(t) \rangle$ is the square-and-average response of the photodetector for the given
lightwave, \( E_R(t) \). This term can also be expressed as the sum of the desired signal and contributions from crosstalk and ASE beat noise components,

\[
R_\lambda \langle E_R^2(t) \rangle = i_s(t) + i_{sx}(t) + i_{ssp}(t) + i_{ss}(t) + i_{sp}(t) + i_{xx}(t) + i_{sps}(t) + i_{xsp}(t) \tag{5.13}
\]

The first term, \( i_s(t) \), represents the desired signal contribution. The beat noise between the signal and crosstalk is denoted by \( i_{sx}(t) \), signal and ASE by \( i_{ssp}(t) \), crosstalk and itself by \( i_{ss}(t) \), ASE and itself by \( i_{sp}(t) \) and crosstalk and ASE by \( i_{xsp}(t) \). Combining all the noise terms, equation (5.13) can be written as

\[
i_p(t) = I_{si} + n_i(t) \tag{5.14}
\]

\[
i_p(t) = R_\lambda P_{sig}(N, \lambda) + n_i(t) \tag{5.15}
\]

The \( i \) subscripts in equations (5.14) and (5.15) can have a value of either '0' or '1', indicating the data bit being received by the photodetector. The summed Gaussian noise components of the combined electrical noise, \( n_i(t) \), can be expressed as

\[
\sigma_i^2 = \sigma_{th}^2 + \sigma_{sh}^2 + \sigma_{sgsp}^2 + \sigma_{sgxt}^2 + \sigma_{spsp}^2 + \sigma_{xtxt}^2 + \sigma_{spxt}^2 \tag{5.16}
\]

where the individual variances are given by

\[
\sigma_{sgsp}^2 = 4RI_{si}P_{ase}B_e/B_o \tag{5.17}
\]

\[
\sigma_{spsp}^2 = I_{si}^2P_{ase}^2B_e(2B_o - B_e)/(P_{sig}B_o)^2 \tag{5.18}
\]
\[ \sigma_{sgxt}^2 = 2 \xi R I_s \text{P}_{xt} \] 
\[ \sigma_{xtxt}^2 = \frac{1}{4} R^2 s_i \sum_{i>j} P_{xt,i} P_{xt,j} \]  
\[ \sigma_{spxt}^2 = \frac{4}{B_o} R^2 B_e \text{P}_{ase} \sum_{i=1}^{N_{xt}} P_{xt,i} \]  
\[ \sigma_{sh}^2 = 2 q R B_e (P_{xt} + \text{P}_{ase} + I_{si}/R) \] 
\[ \sigma_{th}^2 = I_{th}^2 B_e \] 

In the expressions above, \( I_{th} \) is the receiver thermal noise current, \( q \) is the charge of an electron, \( N_{xt} \) is the number of crosstalk components, \( B_e \) is the electronic bandwidth, \( \xi (= 1/2 \text{ see [82]} \) is the polarization mismatch factor between the signal and crosstalk lightwaves and \( P_{xt,i} \) is the crosstalk power of the \( i \)th crosstalk component.

Assuming an ideal extinction ratio and an optimum photocurrent decision threshold for minimizing the BER, the Q factor can be used to provide an estimation of the BER. The expression for the Q factor is given by
\[ Q = \frac{I_1 - I_0}{\sigma_1 + \sigma_0} \] 

where \( \sigma_0 \) and \( \sigma_1 \) represent the noise values (from equation (5.16)) and \( I_0 \) and \( I_0 \) are the received photocurrent values when the received data bit is 0 or 1, respectively. The Q factor is related to the BER through the expression

135
\[ BER = \frac{1}{2} \text{erfc} \left( \frac{Q}{\sqrt{2}} \right) \approx \frac{e^{-Q^2/2}}{Q\sqrt{2\pi}} \]  

(5.25)

where a Q factor of 7 translates to a BER of \(10^{-12}\)

| Table I |
|-----------------|-----------------|
| **System Model Parameters** |
| **Parameter** | **Values** |
| Wavelengths | (1546.99, 1547.80, 1548.60, 1549.40, 1550.20, 1551.00, 1551.80, 1552.60) |
| Electronic Bandwidth \((B_e)\) | 10 GHz |
| Optical Bandwidth \((B_o)\) | 50 GHz |
| Switch element insertion loss \((L_s)\) | 1 dB |
| Waveguide/fiber coupling loss \((L_w)\) | 1 dB |
| Switch loss \((L_{sw})\) | \(2\log_2 NL_s + 4L_w\) dB* |
| Tap loss \((L_{tap})\) | 1 dB |
| Fiber loss \((L_f)\) | 0.2 dB/km |
| Input EDFA Gain \((G_{in})\) | 22 dB |
| Output EDFA Gain \((G_{out})\) | 16 dB, 18 dB, 20 dB |
| Multiplexer loss \((L_{mx})\) | 4 dB |
| Demultiplexer loss \((L_{dm})\) | 4 dB |
| ASE factor \((n_{sp})\) | 1.41 |
| RMS thermal noise current \((I_{th})\) | \(3.8 \times 10^{-12}\) Amp |
| Switch crosstalk ratio \((X_{sw})\) | 20 dB |
| Mux/demux crosstalk ratio \((X_{dm})\) | 30 dB |
| Planck’s constant \((h)\) | \(6.626 \times 10^{-34}\) |
| Rx responsivity \((R)\) | 0.95 Amp/W |
| Max. laser Power \((P_t)\) | 1 mW |
| Q factor threshold \((q_{th})\) | 7 |
5.4 Simulation and Results

We use two well-known networks (shown in Figure 5.7) to illustrate the ideas discussed in this work. The 15N-Mesh network has 15 nodes and 21 links and the NJLATA network has 11 nodes and 23 links. We assume that each link consists of two fibers in opposite directions. We use the link lengths of 100km each, which is consistent with previous work found in the literature [73, 83, 75]. We assume Poisson arrivals of calls (connection requests for lightpaths) with exponential holding times. Connection requests are uniformly distributed among all network node pairs with 1 unit capacity. As shown in the network architecture, we assume a wavelength conversion-free network. Table X lists the parameters used for our simulations.

Our measurements are performed as follows. For each data point in our graphs, we average 100 simulation runs. For each simulation run, we route 20,000 calls for warm-up. Then, we route windows of 5,000 connections. Blocking probabilities in two consecutive windows are compared and the simulation is stopped when
this difference reaches 5% or less. In our experiments, the total number of calls ranged from 40,000 to 120,000 before stopping using this technique. To measure the impact of link failures, after we stop the simulator using the above method, we simulate failures of all links, one by one. We then continue the simulation until all existing connections have departed, and route another 1000 connections before taking measurements again. This process is repeated 50 times for each of the 100 runs. In other words, blocking probability is averaged over 100 runs, and $V_{QoT}$ and $V_{cas}$ are measured over the same 100 runs with 50 measurements within each run.

In performing RWA, we utilize topological shortest paths for primary paths and shortest possible paths that are link-disjoint from the primary paths for backup. All possible shortest primary and backup path pairs are considered.

### 5.4.1 15N-Mesh

Figures 5.8 to 5.9 show blocking probability, $V_{QoT}$ and $V_{cas}$ for DPP and SPP on the 15N-Mesh network. First, the well-known trend between DPP and SPP in terms of blocking performance holds true in AONs, i.e. SPP has significantly lowered blocking across the various algorithms used in this work.

The QoT aware algorithms perform fairly close to the ideal case because the backup paths become “buffers” between primary paths. As a result, wavelength blocking dominates call rejection due to low BER when using QoT-aware algorithms. In DPP, more wavelength channels are reserved for backup, which results in closer to ideal performance compared to SPP. Also note that using DPP, the network can only support up to about 14 Erlangs under the ideal scenario if overall blocking is kept below a reasonable range (about 5%). SPP can support up to about 20 Erlangs.
Figure 5.8: 15N-Mesh DPP.
Figure 5.9: 15N-Mesh SPP.
It is also easy to notice that the QoT aware algorithms perform much better than the QoT unaware algorithms. Also, because the demux/mux crosstalk is the dominating factor (vs. switch crosstalk), there is a bigger merit in using the QoT aware algorithms compared to utilizing the Xsw-free architecture. The blocking probability for QoT-guaranteed protection is considerably higher as expected. When the load is very low however (below 6), the network is empty enough to allow QoT-G algorithm to perform slightly better than the random placement in QoT unaware algorithms.

As the load increases (and more of the network is utilized), QoT Vulnerability for QoT Aware algorithms becomes worse than their unaware counterparts. The two QoT aware algorithms are designed to route a connection by avoiding other primary paths to avoid crosstalk. This procedure places the primary paths alongside/next to either empty channels or backup (non-active) paths. The likelihood of placement next to a backup path increases as the network load increases. The QoT unaware algorithms suffer from high blocking and has less connections sitting on the network at the same load level and our results show that random placement in QoT unaware algorithms performs relatively well.

With cascading failures, QoT aware algorithms perform better than QoT unaware algorithms because the primary paths are selected to minimize (albeit greedily) crosstalk. Upon interference from backup paths, the primary paths provisioned with QoT aware schemes are more likely to survive.

There are two reasons why the \( V_{QoT} \) is much higher than \( V_{casc} \). First, the post-failure BER for primary paths can only get worse by introducing crosstalk from an activated backup path. On the other hand, backup paths can interfere with each other as they are activated. Second, the average path lengths for primary paths is much shorter compared to that of the backup paths (about 2.4 vs. 4.0). Therefore, the BER on primary paths are much lower compared to backup paths,
and the impact of crosstalk is more severe on the backup paths compared to primary paths.

The lower vulnerability in DPP compared to SPP is simply due to the fact that SPP accepts more connections at the same load level and therefore has more potential crosstalk. Given that post-failure BER for backup paths go over the threshold 10 to 40% of the time on average, QoT-guaranteed protection is required to guarantee full link-failure recovery.

5.4.2 NJLATA

Figures 5.10 and 5.11 show the simulation results for NJLATA. The general trend in blocking probability is similar to 15N-Mesh. The performance of QoT-G, however, is much better compared to 15N-Mesh. First, because NJLATA is more well-connected, there are more path choices for each connection making NJLATA more flexible and supportive of alternate routing choices. Second, the average backup path length for NJLATA is around 2.3 compared to 4.0 for 15N-Mesh. The average primary path lengths for NJLATA is around 1.8 compared to 2.4 for 15N-Mesh. Therefore, NJLATA’s primary and backup path pairs are more resilient to signal degradation.

The QoT unaware algorithms outperform QoT aware algorithms in terms of QoT vulnerability for NJLATA. Unlike 15N-Mesh, random placement of paths are more acceptable at low loads due to the stronger SNR in NJLATA. QoT aware algorithms’ lower blocking translates to more connections sitting on the network, but at higher loads, network utilization starts to increase. When the QoT aware algorithms place primary paths away from each other, they are more likely to be sitting next to a backup path causing higher increase in post-failure BER.
Figure 5.10: NJLATA DPP.
Figure 5.11: NJLATA SPP.
Note that $V_{QoT}$ for the various algorithms are closer in SPP compared to DPP. As explained previously, primary paths in QoT aware algorithms are placed next to backup paths to avoid crosstalk during normal operation. When different a set of backup paths are activated (in the event of a link failure), they affect the neighboring primary paths. Under SPP, each wavelength-channel reserved for backup has a higher chance of being activated because they serve multiple connections. Therefore, SPP’s unaffected primary paths and activated backup paths are more likely to hit each other compared to DPP. At the same time, SPP is able to accept more connections at the same load level compared to DPP. Therefore, SPP has a higher level of crosstalk across the network compared to DPP resulting in higher $V_{QoT}$ and $V_{cas}$.

5.4.3 Discussion

If less than, but close to, 100% recovery can be tolerated by the network, the failure ratio metric can be very useful. It essentially quantifies how much rerouting is required. Figures 5.12a and 5.12b show the failure ratios for 15N-Mesh and NJLATA networks. The overall trends are similar to that of $V_{QoT}$. It shows, however, that NJLATA is a good candidate network for utilizing QoT aware algorithms instead of using QoT-G since the failure ratios are below 3% for DPP and below 3.5% for SPP. 15N-Mesh has a high failure ratio and it is safe to assume that rerouting up to 22% of the failed connections is unreasonable, especially in a network that is designed with 100% single failure survivability in mind. Developing rerouting techniques for a small fraction of the connections that fail recovery (as in NJLATA) can be explored, but is outside the scope of this work and left for future work.
Figure 5.12: a) FR for 15N-Mesh DPP (top) & SPP (bottom)  b) FR for NJLATA DPP (top) & SPP (bottom)

5.5 Summary

Transmission impairments such as ASE noise and crosstalk can determine the ultimate performance of AONs. We showed that it also poses additional problems for network survivability.

Simulation results showed that a network designed to survive failures using protection schemes can experience failures with high probability due to a significant drop in the signal to noise ratio. We also showed that cascading failures can occur. Utilizing QoT-guaranteed protection is the only way to insure complete network protection, but suffers from higher call blocking. An alternative approach to minimize this overhead was also presented. Our results showed that, while such algorithms cannot guarantee 100% protection, depending on the topology, they
may be useful if a small number of incomplete recovery is acceptable under the reliability requirements. Various rerouting techniques from the literature can then be applied to reestablish the broken connections.

The results also showed that, unlike previous studies where physical layer constraints were not considered, increasing path lengths for shared protection degrades performance. The length of paths used for backup are critical to network robustness as longer hops lead to higher chance of crosstalk (which in turn affects other connections) in the event of a link failure. This problem is compounded with the fact that such paths have weaker signals due to insufficient compensation from saturated amplifiers as well experiencing additional ASE noise.
In this thesis, several critical areas in the design and management of optical backbone networks with a focus on reliability was studied. As we become increasingly dependent on reliable operation of computer networks, the role of survivable backbone networks will become even more important. Given that many new applications require the flexibility and scalability currently unavailable in the core networks, several new interesting problems that are related to the topics covered in this work arise. First, handling various kinds of failures (such as multiple failures) is becoming more important. For example, future network planning in some cases include the ability to handle three simultaneous or sequential failure scenarios. The ability to efficiently manage network resources while meeting such stringent requirements is challenging and will require better algorithms as well as more efficient planning. To this end, providing cost-efficient, rapid recovery for networks also faces the same problem. We believe that the ability to properly provision the network along with the ability to redimension the network based on traffic migration (or short-term need) will play a critical role in future network management. We have studied such techniques for unprotected scenarios [84], but a better understanding is required to service protected networks as well as all-optical networks with wavelength continuity constraints. Second, IP/WDM architectures relying on differentiated protection mechanisms show the most promise for efficient network utilization. However, the impact of post-failure network performance under such scenarios is not well understood. In [85] we show that rerouting connections
that belong to best-effort class, along with other classes of traffic, may suffer a significant increase in blocking without proper planning. A better understanding of capacity planning is thus required to meet the needs of emerging architectures and to implement various proposed methods for the IP/WDM architecture. Finally, some interesting problems stem from studying the impact of physical layer impairments on reliability. Currently, the benefits of wavelength conversion in terms of general network performance are well-understood. It allows a greater flexibility in path-wavelength selection, but additional impairments can be introduced by the wavelength conversion process. The next logical step is to investigate a model that includes wavelength conversion along with protection algorithms that leverage this flexibility with the goal of achieving maximum network utilization.
REFERENCES


151


Sun-il Kim grew up in Long Island, NY where he split his time between outdoor activities and consuming random information. He graduated from Herricks Junior High School in 1994 and Herricks High School in 1997. Dr. Kim received a B.S. in Computer Science with a minor in Mathematics from the State University of New York at Binghamton in 2000. During his final semester as an undergraduate student, as well as the summer after his graduation, he worked as a software engineer at E-base Inc. in Binghamton, NY.

Dr. Kim moved to Illinois in Fall 2000, and after a short 8 years, has finished his PhD program in the Department of Computer Science while working at the Coordinated Science Laboratory. His doctoral research has focused on management techniques for optical networks.