Survivability in Layered Networks

by

Kayi Lee

M.Eng., Electrical Engineering and Computer Science
MIT, 2000
S.B., Computer Science and Engineering
MIT, 1999
S.B., Mathematics
MIT, 1999

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2011

Author ....
Department of Electrical Engineering and Computer Science
January 10, 2011

Certified by ....
Eytan Modiano
Associate Professor
Thesis Supervisor

Accepted by ....
Terry P. Orlando
Chairman, Department Committee on Graduate Students

Survivability in Layered Networks

by

Kayi Lee

Submitted to the Department of Electrical Engineering and Computer Science
on January 10, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

Abstract

In layered networks, a single failure at the lower (physical) layer may cause multiple failures at the upper (logical) layer. As a result, traditional schemes that protect against single failures may not be effective in layered networks. This thesis studies the problem of maximizing network survivability in the layered setting, with a focus on optimizing the embedding of the logical network onto the physical network.

In the first part of the thesis, we start with an investigation of the fundamental properties of layered networks, and show that basic network connectivity structures, such as cuts, paths and spanning trees, exhibit fundamentally different characteristics from their single-layer counterparts. This leads to our development of a new cross-layer survivability metric that properly quantifies the resilience of the layered network against physical failures. Using this new metric, we design algorithms to embed the logical network onto the physical network based on multi-commodity flows, to maximize the cross-layer survivability.

In the second part of the thesis, we extend our model to a random failure setting and study the *cross-layer reliability* of the networks, defined to be the probability that the upper layer network stays connected under the random failure events. We generalize the classical polynomial expression for network reliability to the layered setting. Using Monte-Carlo techniques, we develop efficient algorithms to compute an approximate polynomial expression for reliability, as a function of the link failure probability. The construction of the polynomial eliminates the need to resample when the cross-layer reliability under different link failure probabilities is assessed. Furthermore, the polynomial expression provides important insight into the connection between the link failure probability, the cross-layer reliability and the structure of a layered network. We show that in general the optimal embedding depends on the link failure probability, and characterize the properties of embeddings that maximize the reliability under different failure probability regimes. Based on these results, we propose new iterative approaches to improve the reliability of the layered networks. We demonstrate via extensive simulations that these new approaches result in embeddings with significantly higher reliability than existing algorithms.

Thesis Supervisor: Eytan Modiano
Title: Associate Professor

Acknowledgments

Writing this thesis would have been impossible without the help and support from many people throughout the years. I would like to take this opportunity to express my gratitude towards everyone who has helped to make this happen.

First and foremost, I would like to thank my thesis supervisor, Professor Eytan Modiano, who is always welcoming to ideas and thoughts, no matter how vague and seemingly inconsequential they are, and has a knack for turning them into inspiring questions. I have benefitted tremendously from Eytan's guidance and support over the years. His insight and enthusiasm in this research area has helped me identify the central theme of this thesis and tackle this problem area from different angles.

I am also greatly indebted to my collaborator, Dr. Hyang-Won Lee, who has contributed significantly in the second part of the thesis. I truly enjoyed the numerous lunch meetings with him at the Google cafeteria, where many results in this thesis originated. Without a doubt those have been the most productive hours for my research.

I would also like to thank my thesis committee members, Professors Dimitri Bertsekas and John Tsitsiklis, for their invaluable comments on the research work. In particular, their suggestion of applying importance sampling to reliability estimation has deepened my understanding of the Monte Carlo techniques, which helped to improve Chapter 3 of this thesis significantly.

I also want to thank Dr. Sunny Siu for advising me during the first few years of my graduate study. He has helped me immensely in my initiation into the area of optical networking and provided invaluable guidance in conducting research in general.

Over the years I have met many wonderful friends at MIT, including my ex-roommate Ching Law, with whom I spent countless hours playing board games, attending classes and discussing research problems; Joyce Ng and Addy Ngan, who are always fun to hang out with; Chung Tin, the nicest guy I have met who is always helpful despite his own busy research schedule; and all the talented friends in the Communications and Networking Research Group. I thank them for all the fond

memories.

A majority of the research in this thesis was conducted on a part-time basis while I was working for Google as a software engineer. I am very grateful to Google for its amazing education program that has allowed me to continue pursuing my research, and its overall culture that encourages learning and professional development. In addition, I must also thank Janet Fischer and Professor Terry Orlando for their administrative and moral support throughout the times I was registered as a part-time student.

Last but not least, I am very thankful to my family for their unwavering support all these years. My parents, Ying Kei Lee and Po Kuen Cheung, have always believed in me and it is their hard work and devotion that have allowed me to receive the best education here at MIT. I am also very fortunate to have my baby girl Natalie to give me a little kick in the final stretch of my thesis writing. My gratitude also extends to my mother-in-law, Ya Sian Yu, for her delicious dishes every day during the challenging times when my wife and I were adjusting to being new parents. Finally, this thesis is dedicated to my wife Sally Kwok, whose patience and positive attitude have provided me with the much-needed energy that carried me through the journey of my PhD study. This thesis is a manifestation of her love and support.

Support

The research conducted in this thesis was supported by NSF grants CNS-0626781 and CNS-0830961, DTRA grants HDTRA1-07-1-0004 and HDTRA-09-1-0050, and the Google Fellowship Program.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Layering is a fundamental concept in modern network design. It describes the decomposition of the network's functions into separate logical components. The way the functions are divided, as well as the interactions among these logical components, define the network architecture. In modern communication networks, these components, called *layers*, are often organized as a stack, where each layer relies on the services provided by the layer below to implement the services used by the layer above. Common network models based on stacked layering include the OSI Reference Model [127] and the TCP/IP model [23]. The decomposition of network functionalities allows each layer to hide much of its internal complexity and provide a clean interface to the client of its services. For instance, in the OSI Reference Model, the physical layer is responsible for providing a "pipe" with a certain amount of bandwidth to the layer above. The actual physical medium that implements the pipe, however, is opaque to the upper layer. Similarly, the data link layer is responsible for framing, multiplexing and demultiplexing data that is sent over the physical layer. It defines the protocol for reliable data transmission over the physical link. This transforms the raw bandwidth provided by the physical layer into channels that allow the upper layer to reliably access and share the physical bandwidth. Such a layering approach greatly simplifies the network design and makes it possible to implement and operate the network in a modularized and evolvable manner.

A pertinent example of a multi-layer network is the IP-over-WDM network, as shown in Figure 1-1. At the lower layer is a Wavelength Division Multiplexing (WDM) network which consists of the optical switches connected by the physical fibers. On top of the WDM network is an IP network where the IP routers are connected using (WDM) *lightpaths.* Each lightpath is realized by setting up a physical connection using one of the wavelength channels in the optical fibers. In this IP-over-WDM architecture, the network topology in the upper layer, called the *logical topology,* is defined by the set of IP routers and the lightpaths connecting them. On the other hand, the *physical topology* is defined by the (possibly different) set of optical switches and the fibers connecting them. In this thesis, we will discuss our results in the context of IP-over-WDM networks; as such, we will use the terms "logical links" and "lightpaths" interchangeably. However, the concepts discussed are equally applicable to other layered architectures, such as IP over ATM, ATM over SONET, etc.

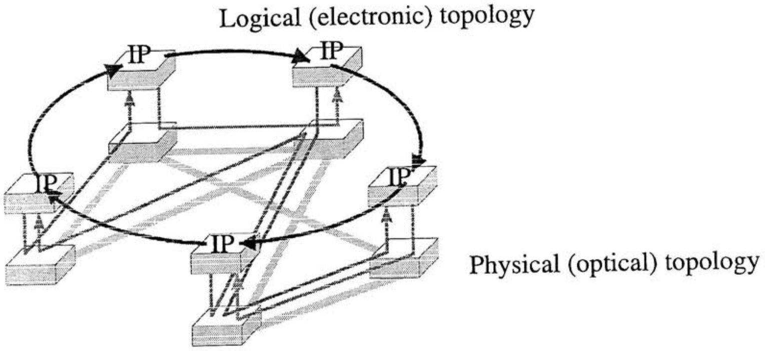
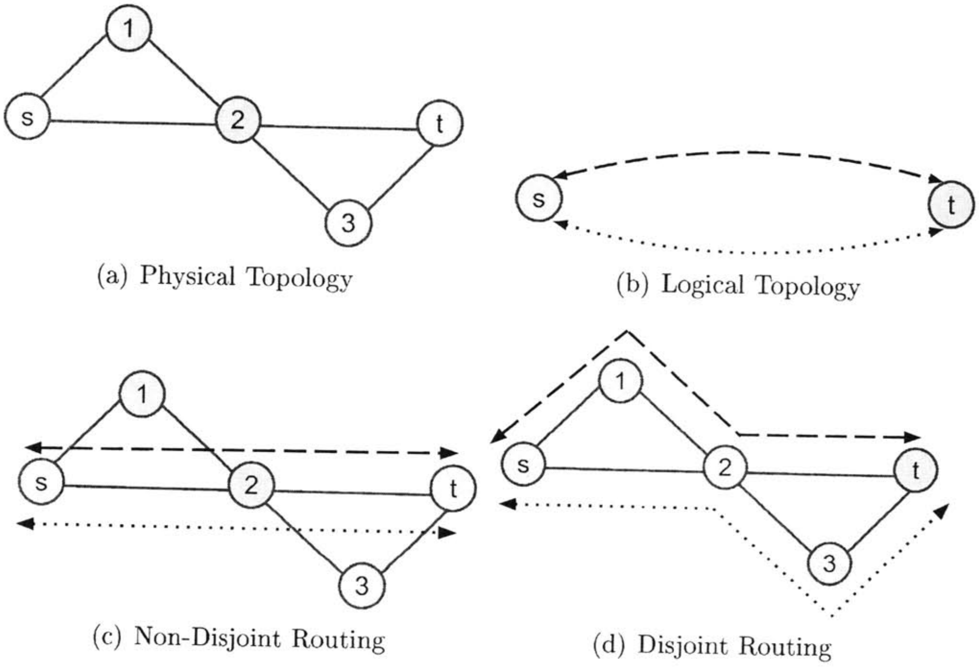


Figure 1-1: An IP-over-WDM network where the IP routers are connected using optical lightpaths. The logical links (arrowed lines on top) are formed using lightpaths (arrowed lines at the bottom) that are routed on the physical fiber (thick gray lines at the bottom). In general, the logical and physical topologies are not the same.

In multi-layer networks, the design of the logical topology is often decoupled from the physical topology. For example, it is very possible that two logical nodes that are connected by a logical link are not directly connected by a physical fiber. In this case, the logical link can be created by setting up a lightpath that traverses multiple physical hops. This, however, involves selecting the physical route taken by the lightpath. The choice of physical routes taken by the lightpaths in the logical topology, called the *lightpath routing,* has significant implication on capacity requirement and network survivability. As an illustrative example, Figures 1-2(a) and 1-2(b) show the physical and logical topologies of a layered network, and Figures 1-2(c) and 1-2(d) show two different lightpath routings. In Figure 1-2(c), the two logical links between s and t are routed over the same physical path. From a capacity standpoint, this means that the physical fiber must have the capacity to support two lightpaths within the same fiber. From a survivability standpoint, this means a single fiber cut can cause both of the logical links to fail simultaneously, thereby disconnecting the logical nodes s and t. As a result, the logical network is susceptible to a single physical failure. In contrast, in Figure 1-2(d), the logical links are routed disjointly over the physical network. In this case, the physical fibers only need the capacity to support one wavelength channel, and any single fiber cut will only result in failure of at most one logical link.



(a) Physical Topology

(b) Logical Topology

(c) Non-Disjoint Routing

(d) Disjoint Routing

Figure 1-2: Routing logical links differently can affect capacity requirement and survivability.

Therefore, by routing the lightpaths intelligently over the physical network, one can increase utilization, as well as improve survivability of the network. While the impact on the utilization has been quite extensively studied [3, 14, 15, 52, 69, 85, 95, 115, 126], the survivability aspect is relatively unexplored. The main focus of this thesis is to develop a deeper understanding on how multi-layer survivability can be achieved by a good lightpath routing. We will consider the following model for a two-layer network:

- A *physical topology* at the lower layer, modelled by a network graph $G_P = (V_P, E_P)$;
- A *logical topology* at the upper layer, modelled by a separate network graph $G_L = (V_L, E_L)$, where $V_L \subseteq V_P$;
- A *lightpath routing*, which maps each logical link $(s,t) \in E_L$ to a physical (s,t)-path in G_P.

Associated with the layered network is a *survivability measure* χ, which maps the lightpath routing to a non-negative real number that quantifies its survivability performance. Throughout the thesis, we will consider different definitions for χ, and study two classes of problems:

1. **Survivability Measurement:** Given the physical and logical topologies, as well as the lightpath routing R as input, compute $\chi(R)$.
2. **Survivable Lightpath Routing:** Given the physical and logical topologies, find the lightpath routing R that maximizes $\chi(R)$.

In the rest of this section, we will provide background on network survivability in Section 1.1, and discuss existing works in cross-layer survivability in Section 1.2. Then in Section 1.3, we will present an outline of the thesis and highlight our major contributions.

1.1 Background on Network Survivability

The two main approaches to providing network survivability are *protection* and *restoration*. Protection refers to rapid and preplanned recovery mechanisms where in the event of a failure, traffic is switched over to back-up paths. On the other hand, restoration refers to recovery mechanisms whereby back-up paths are found dynamically in the event of a failure [50]. Network survivability at a single layer has been studied extensively and the literature on protection and restoration is extremely rich [5, 30, 38, 44, 49–51, 58, 68, 71, 78, 88, 92, 93, 104, 109, 117, 118]. Here we provide a brief overview of protection and restoration in single layer networks; highlighting the issues that are key to this thesis.

Protection can be provided at the various layers [45, 51, 99]. Protection mechanisms are classified into *link protection* and *path protection.* Link protection recovers from a link failure by rerouting the traffic around the failed link (e.g., using loop-back protection [30, 44, 88, 92, 93]). In contrast, path protection reroutes traffic using a back-up end-to-end path for each traffic stream [58, 78, 92, 93, 104]. For example, SONET rings employ either link-based or path-based protection switching [50, 51], to guarantee recovery within 60ms. For path protection, SONET reserves primary and back-up paths in opposite directions around the ring; while link protection is accomplished by rerouting the traffic around the ring from the one end of the failed link to the other [49, 117]. Similarly, both path and link protection can be employed in general mesh network topologies (e.g., ATM, WDM, etc.). Path protection is accomplished by establishing disjoint primary and back-up paths from the source to the destination; where the two paths must be disjoint to ensure that they do not fail simultaneously [92, 93]. Link protection in mesh networks can be accomplished through the use of *protection cycles* that provide a path from the source to the destination of the failed link [38, 104].

In contrast, restoration does not involve preplanning of back-up paths, and is typically provided at the electronic (or logical) layers. The simplest example of restoration is that of packet traffic in the Internet where the Internet Protocol (IP) automatically recovers from link failures by rerouting packets, using its standard routing algorithms (e.g., OSPF, etc.) [57, 68, 71]. Restoration can also be done for connection traffic, on an end-to-end basis; where after a failure, a new path is established dynamically [5, 93, 109]. However, since restoration does not utilize preplanned back-up paths, it typically takes longer to recover from failures. Moreover, failure recovery is not guaranteed as a back-up path may not exist or back-up capacity may not be available.

Different network technologies use either protection or restoration for failure recovery, and the choice is driven by the service being provided. The distinction between protection and restoration is important because they each impose different requirement on the network design. For example, protection is typically done using disjoint primary and back-up paths. Hence network topologies must be able to easily accommodate disjoint paths. For this reason SONET uses a ring architecture where disjoint paths can be easily established around the ring. In contrast, restoration reroutes traffic by finding an alternative path after the failure. This imposes a somewhat less stringent requirement in that the network merely has to remain connected in order to reroute traffic, subject to sufficient capacity.

Typically, protection or restoration is provided at the electronic (logical) layer, because it is needed to recover from electronic layer failures (e.g., line card failure). Although physical layer protection is also possible, it is often very costly in terms of additional protection capacity and is often incompatible with the electronic layer protection mechanism (e.g., SONET protection switching is initiated within a few milliseconds; not nearly enough time for optical layer protection to take effect) [50, 51]. Moreover, since the electronic layer typically offers protection or restoration mechanisms, protection at the physical layer is often redundant [57]. Hence, in this thesis we focus on network architectures where the protection and/or restoration is provided at the electronic layer only.

1.2 Previous Work on Cross-Layer Survivability

While protection and restoration have been extensively studied in single-layer networks, their applicability to cross-layer networks is not well understood. For example, protection mechanisms rely on finding disjoint paths in the network, a well understood problem in single-layer graphs. However, in multi-layer networks, once the logical topology is embedded on the physical topology, a physical fiber link may carry multiple logical links. Therefore, disjoint paths at the logical layer may not be disjoint at the physical layer, rendering the logical layer protection ineffective. Similarly, restoration mechanisms require the network to remain connected after a failure. While connectivity in single-layer graph is well understood, in a multi-layer network, a physical layer failure can lead to multiple logical link failures, which makes it possible to disconnect the logical network even if the logical topology is designed to have high connectivity.

Cross-layer survivability has received relatively limited attention in the literature. Most previous works on cross-layer survivability have been in the context of WDM-based networks and consider very specific objectives, such as routing lightpaths to survive single link failures in optical networks or finding disjoint paths that do not share a common network failure, generally called a Shared Risk Link Group (SRLG) [8,18,19,28,35,37,56,72,84,91,100,103,105,113,120–122,125].

The impact of physical layer failures on the connectivity of the logical topology was first studied by Crochat *et al.* [6,33,34] in the context of WDM-based networks. The authors proposed heuristic algorithms for routing the lightpaths that constitute the logical topology, on the physical topology, so as to minimize the number of disconnected node pairs on the logical topology in an event of single physical link failure. Modiano and Narula-Tam [76] first introduced the notion of Survivable Lightpath Routing, which is defined to be a routing of the logical links over the physical topology so that the logical topology remains connected in the event of a single fiber failure. The same paper developed mathematical conditions for routing lightpath on the physical topology so that the logical topology remains connected even if one of the fibers fails and formulated the problem as an Integer Linear Program (ILP). In [36], Deng, Sasaki and Su developed a Mixed Integer Linear Program (MILP) for the survivable routing problem with polynomial number of constraints. Todimala *et al.* [113] generalized the problem definition to cover single SRLG failures, and developed an ILP as well as heuristic algorithms. The problem of routing logical rings survivably on the physical network was studied in [76,81,101,102]. In particular, [81] considered the physical network design problem and proposed several special physical topologies that guarantee the existence of survivable lightpath routings for logical rings. In [67], Kurant *et al.* introduced the notion of piecewise survivable mapping and developed an algorithm to compute survivable lightpath routings based on piecewise survivable components. The same technique was extended to compute lightpath routings that are survivable against k failures, for a fixed value of k [66]. In [112], Thulasiraman *et al.* introduced the idea of adding *protection edges* to the logical topology in the case where survivable lightpath routing cannot be found by the Kurant's algorithm. Based on this idea, the authors enhanced Kurant's algorithm to always return a survivable lightpath routing, at the expense of the extra protection edges.

The related issue of SRLG failures was introduced in the Generalized Multi-Protocol Label Switching (GMPLS) standard in the IETF for failure management [28, 91, 100]. A SRLG is a group of lightpaths that fail simultaneously upon a single physical failure. For example, for a particular optical fiber, all the lightpaths that traverse the same fiber form a SRLG. Thus, in order to provide rapid protection, two SRLG-disjoint paths, i.e., paths that do not share a common SRLG, must be used. This SRLG-Disjoint Path Problem (SDPP) was first studied in [18] and subsequently in the book written by the same author [19]. In [56] the problem was shown to be NP-complete; and heuristic algorithms for different variations of the SDPP problem were proposed in [8, 72, 84, 103, 120–122]. Various aspects of network design under SRLG constraints were also studied in [35, 37, 105, 113, 125].

1.3 Contributions

1.3.1 Theoretical Underpinnings of Cross-Layer Survivability Problems

As discussed in the previous section, all existing works in cross-layer survivability consider very specific objectives and the primary focus is to design algorithms for these problems. This thesis attempts to develop a more rigorous treatment of cross-layer survivability in order to provide the foundation for quantifying and optimizing survivability in layered networks. We will start with the questions of *why*, and *to what extent*, existing protection and restoration mechanisms do not work in the multi-layer setting. Section 2.2 offers answers to these questions by exposing the structural differences between single-layer and multi-layer networks. More specifically, we propose a model for multi-layer networks that generalizes the classical network graph model for single-layer networks. We will show that connectivity structures in this generalized setting, such as paths, cuts, and spanning trees, exhibit fundamentally different properties from their single-layer counterparts; as such, special graph properties that constitute the foundation of single-layer survivability, such as the max-flow min-cut relationship, do not carry over to multi-layer networks. In addition, we prove several results that reveal the new max-flow min-cut relationship in multi-layered networks, as well as NP-Hardness for computing various basic graph structures in the multi-layer setting, such as maximum disjoint paths, minimum cuts and minimum spanning trees. This collection of results suggest a fundamental structural difference between single-layer and multi-layer networks, which has the following profound implications:

1. Protection and restoration mechanisms designed for single-layer networks may not be effective in the multi-layer setting.

2. Common metrics, such as connectivity, that are used to quantify survivability for single-layer networks lose much of their meanings if applied blindly to multi-layer networks.

3. Existing algorithms for assessing and maximizing survivability for single-layer networks are not easily extendable to the multi-layer setting, due to the fundamental differences between the two types of networks and the inherent hardness of computing multi-layer connectivity structures.

1.3.2 Metrics and Algorithms for Survivable Layered Network Design

The observations from Section 2.2 motivate us to reinvestigate basic issues in survivability for multi-layer networks, starting with the definition of cross-layer survivability. In order to understand the survivability performance of a multi-layer network design, it is important to define metrics that properly capture multi-layer survivability. Unfortunately, due to the inherent complexity of cross-layer structures, defining a meaningful cross-layer survivability metric is non-trivial. Therefore, in Section 2.3 we propose guidelines for cross-layer survivability metric design, defining several properties that a metric must satisfy in order to be a suitable cross-layer survivability metric. Based on these guidelines, we define two cross-layer survivability metrics, called *Min Cross Layer Cut* and *Min Weighted Load Factor.* We will explain their physical meanings and discuss how these metrics can be computed. We will also investigate their mathematical properties, which reveal certain inherent connections between the metrics and provide insight into our development of ILP formulations for the Survivable Lightpath Routing problem.

In Section 2.4 we will formulate the Survivable Lightpath Routing problem as a survivability maximization problem, using Min Cross Layer Cut (MCLC) as the optimization objective. Due to the inherent difficulty in maximizing the metric directly, in Section 2.4 we consider ILP approximations for the MCLC maximization problem. We run extensive simulations comparing the survivability performance of these formulations with the existing Survivable Lightpath Routing algorithm in the literature. The results show that our approach to maximize an approximation of the MCLC can often lead to lightpath routings with significantly better survivability performance than existing algorithms. In addition, our simulation results also suggest that a formulation that closely approximates the MCLC maximization, combined with the randomized rounding technique, provides an efficient way to design multi-layer networks with good survivability performance.

1.3.3 Extension to Random Physical Failures

In the second part of the thesis, we will extend our investigation to the random physical failure model, where all physical links are assumed to fail independently with certain probability. Similar to the deterministic model, a physical link failure will affect all the logical links that use that physical link. The metric of interest under this model is the *cross-layer reliability*, which is the probability that the logical topology stays connected under the random physical failures.

Computing reliability was shown to be $\#P$-complete in single layer networks [114], and even approximating the reliability to within a constant factor cannot be done in polynomial time [87]. Although there are works aimed at exact computation of reliability through graph transformation and reduction [27,73,83,86,98,106,107,111], the applications of such methods are limited to specific topologies. Because of the difficulty in assessing network reliability, most previous works in this context focused on estimating the network reliability, either by deterministic "best-effort" approaches without accuracy guarantee [24,31,53,89,94], or by Monte Carlo simulations [41,62, 63,82] with probabilistic accuracy guarantee.

Although there has been a large body of works on estimating single-layer network reliability, cross-layer reliability has not been explored previously. Our main contributions in this area are new algorithms for cross-layer reliability estimation and maximization, as well as theoretical results that lead to a deeper understanding of structures in layered networks that contribute to high reliability. In Chapter 3, we develop an algorithm that yields a polynomial expression [12] for the reliability of a given multi-layer network. This expression provides a formula for cross-layer reliability as a function of the physical link failure probability. In contrast to many existing reliability estimation methods for single-layer networks [41,62,63], our method is not tailored to a particular probability of link failure, and consequently, it does not require resampling in order to estimate reliability under different values of link failure probability. That is, once the polynomial is estimated, it can be used for any value of link failure probability without resampling.

The polynomial expression given by the algorithm also reveals important structural information of the underlying layered network, which provides clear insights into how lightpath routing should be designed for better reliability. In Chapter 4, we investigate the relationship between the link failure probability, the cross-layer reliability and the structure of a layered network. We show that the structures of the optimal lightpath routings depend on the link failure probability. In particular, lightpath routings that are optimal in the regime where the link failure probability is low, is structurally different from lightpath routings that are optimal in the regime where the link failure probability is high. The investigation culminates in characterizations of optimal lightpath routings in the two probability regimes. These characterizations reveal the criteria for maximizing the cross-layer reliability of lightpath routings under the respective probability regimes, which provides important insights into developing survivable lightpath routing algorithms to maximize cross-layer reliability.

Based on the insights developed in Chapter 4, Chapter 5 explores different methods for maximizing cross-layer reliability of a given lightpath routing in the low probability regime. Specifically, we study two different approaches to improve the reliability of a layered network. The first approach is *lightpath rerouting*, which involves incrementally choosing a new physical route for an existing lightpath, so that the cross-layer reliability can be improved by such a reroute. The second approach is *logical topology augmentation*, where a new lightpath is added to the logical topology to improve reliability. For each approach, we formulate the reliability improvement achieved by a rerouting/augmentation step, and develop algorithms to maximize the reliability improvement. By iteratively applying the algorithm, one can incrementally improve the reliability of the network until no further local improvement is possible. This gives us effective ways to generate lightpath routings with better reliability than all lightpath routing algorithms previously considered. Finally, in Section 5.3, we carry out a case study on a real-world IP-over-WDM network, and apply the techniques discussed in this thesis to study reliability in a real-world setting.

Chapter 2

Fundamentals of Cross-Layer Survivability

2.1 Introduction

A key aspect that is new in the layered network setting is the sharing of physical fibers by multiple logical links. Because of this, a single physical failure will propagate to the logical layer and cause logical links to fail in a correlated fashion. This correlation is implicitly determined by the lightpath routing, and this phenomenon fundamentally changes the connectivity structures of a network. Algorithms designed to effectively assess or enhance survivability of a multi-layer network must therefore take into account such dependencies. Most existing protection and restoration mechanisms for single-layer networks assume uncorrelated failures in the network, and therefore may no longer be effective in this multi-layer setting.

In this chapter, we will develop a more rigorous treatment of fundamental issues in cross-layer survivability. In Section 2.2, we will first study basic connectivity structures, such as cuts, paths and trees, in the multi-layer network model, and highlight the key differences from their single-layer counterparts, both in terms of combinatorial properties and computation complexity. As a result of this, common survivability metrics such as the connectivity of a network topology lose much of their meaning in multi-layer networks. These findings lead us to propose new survivability metrics for multi-layer networks, and algorithms to improve cross-layer survivability based on these new metrics in Sections 2.3 and 2.4. Simulation results for these algorithms will be presented in Section 2.5.

2.2 Graphs Structures in Multi-Layer Networks

In this section, we study various connectivity structures such as flows, cuts, trees and paths in multi-layer graphs in order to develop insights into cross-layer survivability. We will highlight the key difference in combinatorial properties between multi-layer graphs and single-layer graphs. In particular, we will show that fundamental survivability results, such as the Max Flow Min Cut Theorem, are no longer applicable to multi-layer networks. Consequently, metrics such as "connectivity" have significantly different meanings in the cross-layer setting. This motivates our reinvestigation in the following sections of fundamental issues such as quantifying and maximizing survivability in the multi-layer setting.

2.2.1 Max Flow vs Min Cut

For single-layer networks, the Max-Flow Min-Cut Theorem [4] states that the maximum amount of flow passing from the source s to the sink t always equals the minimum capacity that needs to be removed from the network so that no flow can pass from s to t. In addition, if all links have integral capacity, then there exists an integral maximum flow. This implies that the maximum number of disjoint paths between s and t is the same as the minimum cut between the two nodes. Hence, the term *connectivity* between two nodes can be used unambiguously to refer to different measures such as maximum number of disjoint paths or minimum cut, and this makes it a natural choice as the standard metric for measuring network survivability.

Because of its fundamental importance, we would like to investigate the Max-Flow Min-Cut relationship for multi-layer networks. We first generalize the definitions of *Max Flow* and *Min Cut* for layered networks:

Definition 2.1 *In a multi-layer network, the Max Flow between two nodes s and t in the logical topology is the maximum number of physically disjoint $s-t$ paths in the logical topology.*

Definition 2.2 *In a multi-layer network, the Min Cut between two nodes s and t in the logical topology is the minimum number of physical links that need to be removed in order to disconnect the two nodes in the logical topology.*

We model the physical topology as a network graph $G_P = (V_P, E_P)$, where V_P and E_P are the nodes and links in the physical topology. The logical topology is modelled as $G_L = (V_L, E_L)$, where $V_L \subseteq V_P$. The lightpath routing is represented by a set of binary variables f_{ij}^{st}, where a logical link (s,t) uses physical fiber (i,j) if and only if $f_{ij}^{st} = 1$. For any pair of logical nodes x and y, let $\mathcal{P}_{xy}$ be the set of all $x-y$ paths in the logical topology. For each path $p \in \mathcal{P}_{xy}$, let $L(p)$ be the set of physical links used by the logical path p, that is, $L(p) = \cup_{(s,t)\in p} \{(i,j)|f_{ij}^{st} = 1\}$. Then the Max Flow and Min Cut between nodes s and t can be formulated mathematically as follows:

$$\mathsf{MaxFlow_{st}}: \quad \text{Maximize} \sum_{p\in\mathcal{P}_{st}} f_p, \quad \text{subject to:}$$

$$\sum_{p:(i,j)\in L(p)} f_p \leq 1 \qquad \forall (i,j) \in E_P \tag{2.1}$$

$$f_p \in \{0,1\} \qquad \forall p \in \mathcal{P}_{st}$$

$$\mathsf{MinCut_{st}}: \quad \text{Minimize} \sum_{(i,j)\in E_P} y_{ij}, \quad \text{subject to:}$$

$$\sum_{(i,j)\in L(p)} y_{ij} \geq 1 \qquad \forall p \in \mathcal{P}_{st} \tag{2.2}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in E_P$$

The variable f_p in the formulation $\mathsf{MaxFlow_{st}}$ indicates whether the path p is selected for the set of (s,t)-disjoint paths. Constraint (2.1) requires that no selected logical paths share a physical link. Similarly, in the formulation $\mathsf{MinCut}_{\mathsf{st}}$, the variable y_{ij} indicates whether the physical fiber (i, j) is selected for the minimum (s, t)-cut. Constraint (2.2) requires that all logical paths between s and t traverse some physical fiber (i, j) with $y_{ij} = 1$.

Note that the above formulations generalize the the Max Flow and Min Cut for single-layer networks. In particular, the formulations model the classical Max Flow and Min Cut of a graph G if both G_P and G_L are equal to G, and $f_{ij}^{st} = 1$ if and only if $(s, t) = (i, j)$.

Let $\mathsf{MaxFlow}_{\mathsf{st}}$ and $\mathsf{MinCut}_{\mathsf{st}}$ be the optimal values of the above Max Flow and Min Cut formulations. We also denote $\mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}}$ and $\mathsf{MinCut}^{\mathsf{R}}_{\mathsf{st}}$ to be the optimal values to the linear relaxations of above Max Flow and Min Cut formulations. The Max-Flow Min-Cut Theorem for single-layer networks can then be written as follows:

$$\mathsf{MaxFlow}_{\mathsf{st}} = \mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}} = \mathsf{MinCut}^{\mathsf{R}}_{\mathsf{st}} = \mathsf{MinCut}_{\mathsf{st}}.$$

The equality among these values has profound implications on survivable network design for single-layer networks. Because all these survivability measures converge to the same value, it can naturally be used as the standard survivability metric that is applicable to measuring both disjoint paths or minimum cut. Another consequence of this equality is that linear programs (which are polynomial time solvable) can be used to find the minimum cut and disjoint paths in the network.

It is therefore interesting to see whether the same relationship holds for multi-layer networks. First, it is easy to verify that the linear relaxations for the formulations $\mathsf{MaxFlow}_{\mathsf{st}}$ and $\mathsf{MinCut}_{\mathsf{st}}$ maintain a primal-dual relationship, which, by Duality Theorem [17], implies that $\mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}}$=$\mathsf{MinCut}^{\mathsf{R}}_{\mathsf{st}}$. In addition, since any feasible solution to an integer program is also a feasible solution to the linear relaxation, we can establish the following relationship:

Observation 1 $\mathsf{MaxFlow}_{\mathsf{st}} \leq \mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}} = \mathsf{MinCut}^{\mathsf{R}}_{\mathsf{st}} \leq \mathsf{MinCut}_{\mathsf{st}}$.

Therefore, like single-layer networks, the maximum number of disjoint paths between two nodes cannot exceed the minimum cut between them in a multi-layer network.

However, unlike the single-layer case, the values of $\mathsf{MaxFlow}_{\mathsf{st}}$, $\mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}}$ and $\mathsf{MinCut}_{\mathsf{st}}$ are not always identical, as illustrated in the following example. In our examples throughout the section, we use a logical topology with two nodes s and t that are connected by multiple lightpaths. For simplicity of exposition, we omit the complete lightpath routing and only show the physical links that are shared by multiple lightpaths. Theorem 2.1 states that this simplification can be made without loss of generality.

Theorem 2.1 *Let G_L be a logical topology with two nodes s and t, connected by n lightpaths $E_L = \{e_1, e_2, \ldots, e_n\}$, and let $\mathcal{R} = \{R_1, R_2, \ldots, R_k\}$ be a family of subsets of E_L, where each $|R_i| \geq 2$, that captures the fiber-sharing relationship of the logical links. There exist a physical topology $G_P = (V_P, E_P)$ and lightpath routing of G_L over G_P, such that:*

1. *there are exactly k fibers in E_P, denoted by $F = \{f_1, f_2, \ldots, f_k\}$, that are used by multiple lightpaths;*

2. *for each fiber $f_i \in F$, the set of lightpaths using f_i is R_i.*

Proof. See Appendix 2.7.1. □

Theorem 2.1 implies that for a two-node logical topology, any arbitrary fiber-sharing relationship $\mathcal{R}$ can be realized by reconstructing a physical topology and lightpath routing. Therefore, in the following discussion, we can simplify our examples by only giving the fiber-sharing relationship of our two-node logical topology without showing the details of the lightpath routing.

In Figure 2-1, the two nodes in the logical topology are connected by three lightpaths. The logical topology is embedded on the physical topology in such a way that each pair of lightpaths share a fiber. It is easy to see that no single fiber can disconnect the logical topology, and that any pair of fibers would. Hence, the value of $\mathsf{MinCut}_{\mathsf{st}}$ is 2 in this case. On the other hand, the value of $\mathsf{MaxFlow}_{\mathsf{st}}$ is only 1, because any two logical links share some physical fiber, so none of the paths in the logical network are physically disjoint. Finally, the value of $\mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}}$ is 1.5 because a flow of 0.5 can be routed on each of the lightpaths without violating the capacity constraints at the physical layer. Therefore, all three quantities are different in this example. We will study the integrality gaps for the formulations more carefully.

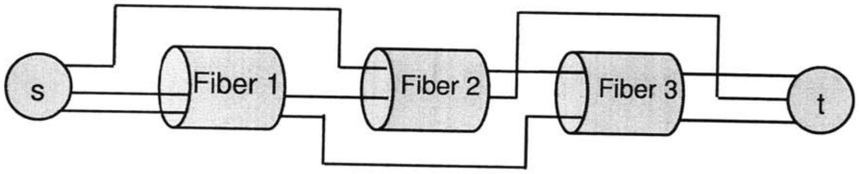


Figure 2-1: A logical topology with 3 links where each pair of links shares a fiber in the physical topology.

Integrality Gap for $\mathsf{MaxFlow}_{\mathsf{st}}$

The above example can be generalized to show that the ratio between $\mathsf{MaxFlow}_{\mathsf{st}}$ and $\mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}}$ is $O(n)$, where n is the number of paths between s and t. Consider an instance of lightpath routing where the two nodes in the logical network are connected by n logical links, and every pair of logical links share a separate fiber. In this case, the value of $\mathsf{MaxFlow}_{\mathsf{st}}$ will be 1, and the value of $\mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}}$ will be $\frac{n}{2}$, using the same arguments as above. Therefore, the ratio $\frac{\mathsf{MaxFlow}_{\mathsf{st}}}{\mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}}}$ is $O(n)$. Note that this is an asymptotically tight bound since $\mathsf{MaxFlow}_{\mathsf{st}} \geq 1$ and $\mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}} \leq n$ for all lightpath routings.

Integrality Gap for $\mathsf{MinCut}_{\mathsf{st}}$

The ratio between $\mathsf{MinCut}_{\mathsf{st}}$ and $\mathsf{MinCut}^{\mathsf{R}}_{\mathsf{st}}$ can be shown to be at most $O(\log n)$ as a direct application of the result by Lovász [74], who showed that the integrality gap between integral and fractional set cover is $O(\log n)$. We can construct a lightpath routing where the gap between the two values is $O(\log n)$, thereby showing the tightness of the bound.

Consider a layered network consisting of a two-node logical topology, and a set of

k fibers $F = \{f_1, \ldots, f_k\}$ that are shared by multiple logical links. For every subset T of $\lfloor \frac{k}{2} \rfloor + 1$ fibers in F, we add a logical link between the two logical nodes that uses only the fibers in T. Hence, for every set of $\lceil \frac{k}{2} \rceil - 1$ fibers, there is a logical link that does not use any of the fibers. This implies the Min Cut is at least $\lceil \frac{k}{2} \rceil$.

On the other hand, since each logical link uses exactly $\lfloor \frac{k}{2} \rfloor + 1$ fibers, the assignment where each $y_{ij} = \frac{1}{\lfloor \frac{k}{2} \rfloor + 1}$ satisfies Constraint (2.2), and is therefore a feasible solution to $\mathsf{MinCut}^{\mathsf{R}}_{\mathsf{st}}$. The objective value of this solution is $\frac{k}{\lfloor \frac{k}{2} \rfloor + 1}$, which is at most 2. Therefore, the integrality gap $\frac{\mathsf{MinCut}_{\mathsf{st}}}{\mathsf{MinCut}^{\mathsf{R}}_{\mathsf{st}}}$ is at least $\frac{k}{4}$.

Therefore, for the two-node logical network with $n = \binom{k}{\lfloor \frac{k}{2} \rfloor + 1}$ logical links, the ratio between the integral and relaxed optimal values for the Min Cut is $O(k) = O(\log n)$. We summarize our observation as follows:

Observation 2 *In a layered network, the values of* $\mathsf{MaxFlow}_{\mathsf{st}}$, $\mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}}$ *and* $\mathsf{MinCut}_{\mathsf{st}}$ *can be all different. In addition, the gaps among the three values are not bounded by any constant.*

Therefore, a multi-layer network with high connectivity value (i.e. that tolerates a large number of failures) does not guarantee existence of physically disjoint paths. This is in sharp contrast to single-layer networks where the number of disjoint paths is always equal to the minimum cut.

It is thus clear that network survivability metrics across layers are not trivial extensions of the single layer metrics. New metrics need to be carefully defined in order to measure cross-layer survivability in a meaningful manner. In Section 2.3, we will specify the requirements for cross-layer survivability metrics, and propose two new metrics that can be used to measure the connectivity of multi-layer networks.

2.2.2 Minimum Survivable Path Set

In this section, we introduce another graph structure, called *Survivable Path Set*, that is useful in describing connectivity in layered networks. A survivable path set for two logical nodes s and t is a set of $s - t$ logical paths such that at least one of the paths in the set survives for any single physical link failure. The Minimum Survivable Path Set, denoted as $\mathsf{MinSPS_{st}}$, is the size of the smallest survivable path set. For convenience, $\mathsf{MinSPS_{st}}$ is defined to be ∞ if no survivable path set exists.

In a single layer network, the value of $\mathsf{MinSPS_{st}}$ reveals nothing more than the existence of disjoint paths, as its value is either 2 or ∞, depending on whether disjoint paths between s and t exist. However, for multi-layer networks, $\mathsf{MinSPS_{st}}$ can be any integer between 2 and ∞[1]. For example, in Figure 2-1, the minimum survivable path set for s and t has size three because any pair of logical links can be disconnected by a single fiber failure. In fact, it is easy to verify that:

- $\mathsf{MinSPS_{st}} = 2$ if and only if $\mathsf{MaxFlow_{st}} \geq 2$;
- $\mathsf{MinSPS_{st}} = \infty$ if and only if $\mathsf{MinCut_{st}} = 1$.

Therefore, the value of $\mathsf{MinSPS_{st}}$ provides a different perspective about the connectivity between two nodes in the cross-layer setting. It is particularly interesting in the regime where $\mathsf{MaxFlow_{st}} = 1$ and $\mathsf{MinCut_{st}} \geq 2$, i.e., there is a gap between the Max Flow and the Min Cut. The following theorem reveals a connection between survivable path sets and the relaxed Max Flow $\mathsf{MaxFlow^R_{st}}$.

Theorem 2.2 $\mathsf{MinSPS_{st}} \leq \left\lfloor \frac{\log |E_P|}{\log \mathsf{MaxFlow^R_{st}}} \right\rfloor + 1.$

Proof. See Appendix 2.7.2 □

It is worth noting that the theorem provides a sufficient condition for the existence of disjoint paths in the layered networks, in terms of the optimal value of $\mathsf{MaxFlow^R_{st}}$:

Corollary 2.3 *Disjoint paths between two nodes s and t exist in a layered network if the relaxed Max Flow,* $\mathsf{MaxFlow^R_{st}}$, *is greater than* $\sqrt{|E_P|}$.

Proof. By Theorem 2.2, a survivable path set of size two exists if $\mathsf{MaxFlow^R_{st}} > \sqrt{|E_P|}$. This implies the existence of $s - t$ disjoint paths in the layered network. □

[1] An instance with $\mathsf{MinSPS_{st}} = k$ can be easily constructed using the 2-node, k-link logical topology similar to Figure 2-1, in which every set of $k - 1$ logical links share a common physical fiber.

Therefore, survivable path sets not only are interesting graph structures that describe connectivity of layered networks, they can also be useful in revealing the relationship between integral and fractional flows in the layered network.

2.2.3 Spanning Trees

For a single-layer graph $G = (V, E)$, a spanning tree can be defined as a minimal set of edges in E that keeps all nodes in V connected. Since all spanning trees of the graph have the same number of edges, constructing, counting and sampling spanning trees in a single-layer network can be done in polynomial time [46,47,61,82,96]. These nice properties about spanning trees in single-layer networks allow construction of efficient algorithms for reliable single-layer networks design [39,82,108].

For multi-layer networks, however, the characteristics of spanning trees is vastly different. We define a *cross-layer spanning tree* as follows:

Definition 2.3 *In a multi-layer network, a Cross-Layer Spanning Tree is a minimal set of physical fibers whose survival will keep the logical topology connected.*

Unlike single-layer networks, the number of edges in a cross-layer spanning trees can vary significantly. Consider Figure 2-2, which shows the lightpath routing of a two-node logical topology over the physical network with three links. In the example, $\{1, 2\}$ and $\{3\}$ are two minimal sets of physical links that keep the logical topology connected. Therefore, not all cross-layer spanning trees have the same cardinality. In fact, the example can be easily modified such that one of the logical links traverses an arbitrary number of physical fibers. This means that cross-layer spanning trees in a multi-layer network can have significantly different sizes.

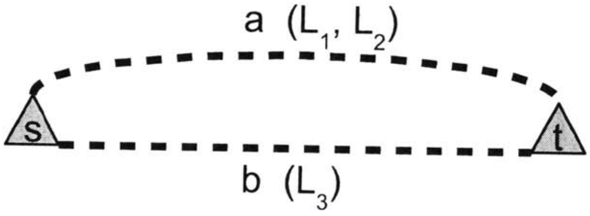


Figure 2-2: $\{L_1, L_2\}$ and $\{L_3\}$ are cross-layer spanning trees with different cardinalities.

The *minimum cross-layer spanning tree* of a layered network, defined to be the cross-layer spanning tree with the minimum number of physical fibers, is of particular importance for cross-layer survivability. Intuitively, this is the minimum number of physical fibers that need to survive in order to keep the logical topology connected. In Chapter 4, we will investigate in greater details the role of minimum cross-layer spanning trees in cross-layer survivability. The following theorem gives a lower bound on the size of the minimum cross-layer spanning tree in a network:

Theorem 2.4 *The size of the minimum cross-layer spanning tree is at least $|V_L| - 1$, where V_L is the set of the logical nodes.*

Proof. For a set of physical links S to be a cross-layer spanning tree, all nodes in V_L must be connected in the underlying physical subgraph induced by S. For S to span a set of $|V_L|$ nodes, it must contains at least $|V_L| - 1$ edges. □

2.2.4 Computational Complexity

The structures discussed in the previous sections are basic building blocks for many survivability algorithms for single layer networks [4, 39, 43, 62, 82, 108]. These algorithms are effective for single-layer networks because these basic structures can be computed efficiently. However, in multi-layer networks, such structures become significantly more difficult to compute, making network survivability measurement and design much more difficult in the multi-layer setting. In this section, we will prove several complexity results for the graph structures introduced in the previous sections.

Max Flow and Min Cut

For single-layer networks, because the integral Max Flow and Min Cut values are always identical to the optimal relaxed solutions, these values can be computed in polynomial time [4]. However, computing and approximating their cross-layer equivalents turns out to be much more difficult. Theorem 2.5 describes the complexity of computing the Max Flow and Min Cut for multi-layer networks.

Theorem 2.5 *Computing Max Flow and Min Cut for multi-layer networks is NP-hard. In addition, both values cannot be approximated within any constant factor, unless P=NP.*

Proof. The Max Flow can be reduced from the NP-hard Maximum Set Packing problem [48]:

Maximum Set Packing: *Given a set of elements $E = \{e_1, e_2, \ldots, e_n\}$ and a family $\mathcal{F} = \{C_1, C_2, \ldots, C_m\}$ of subsets of E, find the maximum value k such that there exist k subsets $\{C_{j_1}, C_{j_2}, \ldots, C_{j_k}\} \subseteq \mathcal{F}$ that are mutually disjoint.*

Given an instance of Maximum Set Packing, we construct a 2-node logical topology connected by multiple lightpaths as described in Theorem 2.1, so that the optimal value of the Maximum Set Packing instance equals the maximum number of physically disjoint paths in the 2-node logical topology. This means that Maximum Set Packing is polynomial time reducible to the 2-node disjoint path problem. Theorem 2.1 implies that any instance of the 2-node disjoint path problem is polynomial time reducible to an instance of the multi-layer Max Flow problem. It follows that Maximum Set Packing is polynomial time reducible to the multi-layer Max Flow problem. Therefore, computing the multi-layer Max Flow is NP-Hard.

Given an instance of Maximum Set Packing with ground set E and a family $\mathcal{F}$ of subsets of E, we construct a logical topology with two nodes, s and t, connected by $|\mathcal{F}|$ logical links, where each logical link corresponds to a subset in $\mathcal{F}$. The logical links are embedded on the physical network in a way that two logical links share a physical fiber if and only if their corresponding subsets share a common element in the Maximum Set Packing instance. It immediately follows that a set of physically disjoint $s-t$ paths in the logical topology corresponds to a family of mutually disjoint subsets of E.

Similarly, the Min Cut can be reduced from the NP-hard Minimum Set Cover problem [48]:

Minimum Set Cover: *Given a set $E = \{e_1, e_2, \ldots, e_n\}$ and a family $\mathcal{F} =$*

$\{C_1, C_2, \ldots, C_m\}$ of subsets of E, find the minimum value k such that there exist k subsets $\{C_{j_1}, C_{j_2}, \ldots, C_{j_k}\} \subseteq \mathcal{F}$ that cover E, i.e., $\bigcup_{i=1,\ldots,k} C_{j_i} = E$.

Given an instance of Minimum Set Cover with ground set E and family of subsets $\mathcal{F}$, we construct a logical topology that contains two nodes connected by a set of $|E|$ logical links, where each logical link l_i corresponds to the element e_i. The logical links are embedded on the physical network in a way that exactly $|\mathcal{F}|$ fibers, namely $\{f_1, \ldots, f_{|\mathcal{F}|}\}$, are used by multiple logical links, and the logical link l_i uses physical fiber f_j if and only if $e_i \in C_j$. It follows that the minimum number of physical fibers that forms a cut between the two logical nodes equals the size of a minimum set cover.

The inapproximability result follows immediately from the inapproximabilities of the Maximum Set Packing and Minimum Set Cover problems [11, 54, 75]. □

Minimum Survivable Path Set

As discussed in Section 2.2.2, the size of Minimum Survivable Path Set for single-layer networks is either 2 or ∞, depending on whether the network graph is bi-connected. Therefore, the Minimum Survivable Path Set can be easily computed in single-layered networks. In multi-layer networks, the Minimum Survivable Path Set can take on many different sizes, and computing its value becomes NP-Hard and inapproximable, just like the cross-layer Max Flow and Min Cut:

Theorem 2.6 *Computing Minimum Survivable Path Set for multi-layer networks is NP-hard. In addition, it cannot be approximated within any constant factor, unless P=NP.*

Proof. The NP-Hardness for the Minimum Survivable Path Set problem can be proved by a reduction from the **Minimum Set Cover** problem similar to Theorem 2.5. Given an instance of Minimum Set Cover with ground set E and family of subsets $\mathcal{F}$, we construct a logical topology that contains two nodes connected by a set of $|\mathcal{F}|$ logical links, where each logical link l_i corresponds to the set $C_i \in \mathcal{F}$. The logical links are embedded on the physical network in a way that exactly $|E|$ fibers, namely $\{f_1, \ldots, f_{|E|}\}$, are used by multiple logical links, and the logical link l_i uses physical fiber f_j if and only if $e_j \notin C_i$. In this case, a set of logical links form a survivable path set between s and t if and only if, for any fiber f_j, there exists a logical link l_i in the path set that does not use f_j. This implies element e_j is covered by the set C_i in the corresponding Minimum Set Cover instance. This proves the NP-Hardness and inapproximity of Minimum Survivable Path Set. □

Minimum Spanning Tree

Since all spanning trees in a single-layer network have the same number of edges, computing a minimum spanning tree is trivial. In multi-layer networks, finding a minimum (cadinality) spanning tree becomes an intractable problem, as described in Theorem 2.7:

Theorem 2.7 *Given the lightpath routing for a multi-layer network $\mathcal{G} = (G_P, G_L)$, finding its Minimum Cross-Layer Spanning Tree is NP-hard.*

Proof. We prove the theorem by constructing a reduction from the NP-Hard *Minimum Label Spanning Tree* problem [26]:

Minimum Label Spanning Tree: *Given a graph $G = (V, E)$, and a set of labels $\mathcal{L} = \{L_1, \dots, L_m\}$. Each edge $e \in E$ is associated with a set of labels $\mathcal{L}_e \subseteq \mathcal{L}$. Find a spanning tree T of G with minimum number of labels, that is, the value $| \cup_{e \in T} \mathcal{L}_e|$ is minimized.*

Given an instance of the Minimum Label Spanning Tree problem, we will construct an instance of the Minimum Cross-Layer Spanning Tree problem, such that the optimal value of the two instances are preserved. The details of the reduction are described in Appendix 2.7.3. □

In summary, multi-layer connectivity exhibits fundamentally different structural properties from its single-layer counterpart. Because of that, it is important to reinvestigate issues of quantifying, measuring as well as optimizing survivability in multi-layer networks. In the rest of the chapter, we will focus on designing appropriate metrics for layered networks, and developing algorithms to maximize the cross-layer survivability.

2.3 Metrics for Cross-Layer Survivability

The previous section demonstrates the new challenges in designing survivable layered network architectures. Insights into quantifying and optimizing survivability are fundamentally different between the single-layer and multi-layer settings. In this section, we focus on the issue of quantifying survivability in multi-layer networks. Not only should such metrics have natural physical meaning in the cross-layer setting, they should also be mathematically consistent and compatible with the conventional single-layer connectivity metric. Hence, we first define formal requirements for metrics that can be used to quantify cross-layer survivability:

- **Consistency:** A network with a higher metric value should be more resilient to failures.

- **Monotonicity:** Any addition of physical or logical links to the network should not decrease the metric value.

- **Compatibility:** The metric should generalize the connectivity metric for single-layer networks. In particular, when applied to the degenerated case where the physical and logical topologies are identical, the metric should be equivalent to the connectivity of the topology.

A metric that carries all the above properties would give us a meaningful and consistent measure of survivability in the multi-layer setting. We propose two metrics, the *Min Cross Layer Cut* and the *Weighted Load Factor*, that can be used to quantify survivability for multi-layer networks. It is easy to verify that both metrics satisfy the above requirements.

2.3.1 Min Cross Layer Cut

In Section 2.2, we defined $\mathsf{MinCut_{st}}$ to be the minimum number of physical failures that would disconnect logical nodes s and t. One can easily generalize this by taking the minimum over all possible node pairs to obtain a global connectivity metric. We define the *Min Cross Layer Cut (MCLC)* to be the minimum number of physical failures that would disconnect the logical topology.

A lightpath routing with high Min Cross Layer Cut value implies that the network remains connected even after a large number of physical failures. It is also a generalization of the survivable lightpath routing definition in [76], since a lightpath routing is survivable if and only if its Min Cross Layer Cut is greater than 1.

Let S be a subset of the logical nodes V_L, and $\delta(S)$ be the set of the logical links with exactly one end point in S. Let H_S be the minimum number of physical links failures required to disconnect all links in $\delta(S)$. The Min Cross Layer Cut can be defined as follows:

$$MCLC = \min_{S \subset V_L} H_S.$$

For each S, computing H_S can be considered as finding the Min Cut between the two partitions S and $V_L - S$. In the proof of Theorem 2.5, we have shown that computing the value of $\mathsf{MinCut_{st}}$ is NP-Hard even if the logical topology contains just two nodes. This immediately implies that computing the global MCLC value is NP-Hard:

Theorem 2.8 *Computing the MCLC for a layered network is NP-Hard.*

In practice, however, the MCLC is bounded by the node degree of the logical topology, which is usually a small constant d. In that case, the MCLC can be computed in polynomial time by enumerating all physical fiber sets with up to d fibers. To compute the MCLC of a layered network in a general setting, it can be modelled by the following integer linear program.

Given the physical and logical topologies (V_P, E_P), and (V_L, E_L), let f_{ij}^{st} be binary constants that represent the lightpath routing, such that logical link (s,t) uses physical fiber (i,j) if and only if $f_{ij}^{st} = 1$. The *MCLC* can be formulated as the integer program below:

$$\mathsf{M}_{\mathsf{MCLC}} : \text{Minimize} \sum_{(i,j)\in E_P} y_{ij}, \quad \text{subject to:}$$

$$d_t - d_s \leq \sum_{(i,j)\in E_P} y_{ij} f_{ij}^{st} \qquad \forall (s,t) \in E_L \tag{2.3}$$

$$\sum_{n\in V_L} d_n \geq 1, d_0 = 0 \tag{2.4}$$

$$d_n, y_{ij} \in \{0,1\} \qquad \forall n \in V_L, (i,j) \in E_P$$

The integer program contains a variable y_{ij} for each physical link (i,j), and a variable d_k for each logical node k. Constraint (2.3) maintains the following property for any feasible solution: if $d_k = 1$, the node k will be disconnected from node 0 after all physical links (i,j) with $y_{ij} = 1$ are removed. To see this, note that since $d_k = 1$ and $d_0 = 0$, any logical path from node 0 to node k contains a logical link (s,t) where $d_s = 0$ and $d_t = 1$. Constraint (2.3) requires that such a logical link traverse at least one of the fibers (i,j) with $y_{ij} = 1$. As a result, all paths from node 0 to node k must traverse one of these fibers, and node k will be disconnected from node 0 if these fibers are removed from the network. Constraint (2.4) requires node 0 to be disconnected from at least one node, which ensures that the set of fibers (i,j) with $y_{ij} = 1$ forms a global Cross Layer Cut.

In Section 2.4, we will use MCLC as the objective for the survivable lightpath routing problem, and develop algorithms to maximize this objective.

2.3.2 Weighted Load Factor

Another way to measure the connectivity of a layered network is by quantifying the "impact" of each physical failure. The *Weighted Load Factor (WLF)*, an extension of the metric *Load Factor* introduced in [60], provides such a measure of survivability.

Given the physical topology (V_P, E_P) and logical topology (V_L, E_L), let f_{ij}^{st} be binary constants that represent the lightpath routing, such that logical link (s,t) uses physical fiber (i,j) if and only if $f_{ij}^{st} = 1$. The WLF can be formulated as follows:

$$\mathsf{M}_{\mathsf{WLF}} : \text{Maximize } \frac{1}{z}, \quad \text{subject to:}$$

$$z \cdot \sum_{(s,t)\in\delta(S)} w_{st} \geq \sum_{(s,t)\in\delta(S)} w_{st} f_{ij}^{st} \qquad \forall S \subset V_L, (i,j) \in E_P$$

$$\sum_{(s,t)\in\delta(S)} w_{st} > 0 \qquad \forall S \subset V_L$$

$$0 \leq z, w_{st} \leq 1 \qquad \forall (s,t) \in E_L,$$

where $\delta(S)$ is the cut set of S, i.e., the set of logical links that have exactly one end point in S.

The variables w_{st} are the weights assigned to the lightpaths. Over all possible logical cuts, the variable z measures the maximum fraction of weight inside a cut carried by a fiber. Intuitively, if we interpret the weight to be the amount of traffic in the lightpath, the value z can be interpreted as the maximum fraction of traffic across a set of nodes disrupted by a single fiber cut. The Weighted Load Factor formulation, defined to maximize the reciprocal of this fraction, thus tries to compute the logical edge weights that minimize the maximum fraction. This effectively measures the best way of spreading the weight across the fibers for the given lightpath routing. A lightpath routing with a larger Weighted Load Factor value means that it is more capable of spreading its weight within any cut across the fibers.

The Weighted Load Factor also generalizes the survivable lightpath routing defined in [76], since its value will be greater than 1 if and only if the lightpath routing is survivable.

Although the formulation $\mathsf{M}_{\mathsf{WLF}}$ contains the quadratic terms zw_{st}, the optimal value of z can be obtained by iteratively solving the linear program with different fixed values of z. Using binary search over the range of z, we can find the minimum z where a feasible solution exists.

Computing the Weighted Load Factor is easier than computing MCLC in certain cases. For example, when the logical topology contains only two nodes with multiple logical links between them, finding the Weighted Load Factor can be formulated as a linear optimization problem:

$$\begin{aligned} \text{Maximize} \quad & \sum_{(s,t)\in E_L} w_{st}, \quad \text{subject to:} \\ \sum_{(s,t)\in E_L} w_{st} f_{ij}^{st} \leq 1 \quad & \quad \forall (i,j) \in E_P, \\ 0 \leq w_{st} \leq 1 \quad & \quad \forall (s,t) \in E_L, \end{aligned}$$

by replacing $\frac{1}{z}$ in the formulation $\mathsf{M}_{\mathsf{WLF}}$ by $\sum_{(s,t)\in E_L} w_{st}$. It can be easily verified that the two formulations are equivalent when the logical topology contains only two nodes.

Therefore, for certain special cases such as the two node logical network, computing the Weighted Load Factor appears to be easier than Min Cross Layer Cut. However, in general, the formulation $\mathsf{M}_{\mathsf{WLF}}$ contains an exponential number of constraints, and may not be polynomial time solvable. In fact, Theorem 2.9 states that finding the objective value for $\mathsf{M}_{\mathsf{WLF}}$ is NP-Hard, even if the weights of the logical links w_{st} are given.

Theorem 2.9 *Computing the Weighted Load Factor for a lightpath routing is NP-Hard even if the weight assignment w_{st} for the logical links is fixed.*

Proof. The NP-Hardness proof is based on the reduction from the NP-Hard *Uniform Sparsest Cut [7]* problem. For details, see Appendix 2.7.4. □

Finally, Theorem 2.10 describes the relationship between the WLF and the MCLC. Given a lightpath routing, let $\mathsf{M}_{\mathsf{MCLC}}$ be the ILP formation for its Min Cross Layer Cut, and let $MCLC$ and $MCLC^R$ be the optimal values for $\mathsf{M}_{\mathsf{MCLC}}$ and its linear relaxation respectively. In addition, let WLF be the Weighted Load Factor of the lightpath routing. Then we have the following relationship:

Theorem 2.10 $MCLC^R \leq WLF \leq MCLC.$

Proof. See Appendix 2.7.5. □

Therefore, although the two metrics appear to measure different aspects of network connectivity, they are inherently related. In fact, as we will see in Section 2.5, the two values are often identical. The connection between the two metrics thus provides insights into the development of the lightpath routing formulation $\mathsf{MCF_{LF}}$, to be introduced in Section 2.4.2.

As a concluding remark of this section. The two metrics introduced in this section are both NP-hard to compute. It remains an interesting open question whether any meaningful cross-layer survivability metrics that is polynomial time computable exists.

2.4 Lightpath Routing Algorithms for Maximizing MCLC

In this section, we consider the survivable lightpath routing problem using the Min Cross Layer Cut as the objective. At an abstract level, the optimal lightpath routing can be expressed as the following optimization problem:

$$\max_{f \in \mathcal{F}} \min_{S \subset V_L} MFC(f, S),$$

where $\mathcal{F}$ is set of all possible lightpath routings, V_L is the logical node set, and $MFC(f, S)$ is the minimum number of fibers whose removal will disconnect all logical links in the cut set $\delta(S)$ given the lightpath routing f. This is a Max-Min-Min problem that may not have a simple formulation. In Section 2.4.1, we first present an ILP formulation that maximizes the MCLC for the lightpath routing. However, the formulation has a large number of variables and is diffcult to solve in practice. Therefore, in Section 2.4.2 we will present several simpler formulations that approximate MCLC maximization.

2.4.1 ILP for MCLC Maximization

We first present a survivable lightpath routing ILP that maximizes the MCLC value:

1. Parameters:

   - $G_P = (V_P, E_P)$: Physical topology.
   - $G_L = (V_L, E_L)$: Logical topology.
   - d: The minimum cut of the logical topology.
   - $\mathcal{C}$: The family of all possible subsets of physical fibers with size at most d.
   - W_i: A weight associated to each fiber set with size i:

$$W_i = \begin{cases} 1, & \text{if } i = |E_P|, \\ 1 + \sum\limits_{i+1}^{|E_P|} \binom{|E_P|}{k} W_k, & \text{if } 1 \leq i \leq |E_P| - 1. \end{cases}$$

2. Variables:

   - $f_{ij}^{st} \in \{0, 1\}$ for $(s, t) \in E_L, (i, j) \in E_P$: Represents the lightpath routing, where $f_{ij}^{st} = 1$ if and only if logical link (s, t) uses fiber (i, j).
   - $y_C \in [0, 1]$ for $C \in \mathcal{C}$: Represents whether the fiber set C is a cross-layer cut. The fiber set C is a cross-layer cut if and only if its value is 1.
   - $x_{st}^{v,C} \in [0, 1]$, for $(s, t) \in E_L, v \in V_L - \{0\}, C \in \mathcal{C}$: Flow variable on the surviving logical topology when fibers in C fail. This is used to express the connectedness of the surviving logical topology under this set of physical failures.

3. Formulation:

$$\mathsf{MCLC\_MAX}: \quad \text{Minimize} \sum_{i=1}^{d} W_i \sum_{C \in \mathcal{C}:|C|=i} y_C, \quad \text{subject to:}$$

$$x_{st}^{v,C} \leq 1 - f_{ij}^{st}, \qquad \forall (i,j) \in C, (s,t) \in E_L, v \in V_L - \{0\}, C \in \mathcal{C} \tag{2.5}$$

$$\sum_{t:(s,t)\in E_L} x_{st}^{v,C} - \sum_{t:(t,s)\in E_L} x_{ts}^{v,C} = \begin{cases} 1 - y_C, \text{ if } s = 0 \\ y_C - 1, \text{ if } s = v, \\ 0, \text{ otherwise.} \end{cases} \qquad \forall v \in V_L - \{0\}, C \in \mathcal{C} \tag{2.6}$$

$$\{f_{ij}^{st} : (i,j) \in E_P\} \text{ forms an } (s,t)\text{-path}, \ \forall (s,t) \in E_L$$

$$f_{ij}^{st} \in \{0,1\}, \ x_{st}^{v,C} \geq 0, \ 0 \leq y_C \leq 1.$$

The objective of the formulation is to minimize the total weighted sum of the cross-layer cuts. Since W_i is defined in a way that the weight of a cross-layer cut with size i dominates the total weights of all cross-layer cuts with size greater than i, the formulation will avoid creating a lightpath routing with small cross-layer cuts. As a result, the optimal solution will have a maximum MCLC value. In addition, since the connectivity of the logical topology is d, the MCLC value of any lightpath routing is at most d. Therefore, it is sufficient to have the objective consider physical fiber sets with size up to d.

By Constraints (2.5) and (2.6), the variable $x_{st}^{v,C}$ represents the amount of flow sent from logical node 0 to node v along the logical link (s,t), under the scenario where fibers in C fail, causing all logical links that use these fibers to fail. Specifically, Constraint (2.5) makes sure that a positive flow can be assigned to logical link (s,t) only if the logical link (s,t) does not use any of the physical fiber $(i,j) \in C$. In other words, only the surviving logical links under the failure event C can be used. Constraint (2.6) is the flow conservation constraint on the logical topology with flow value $1 - y_C$. If the logical topology remains connected under the failure event C, a positive flow can be sent from node 0 to any other node v, and y_C can therefore be set to 0. On the other hand, if the logical topology is disconnected, node 0 will be disconnected to some logical node v, in which case y_C has to be set to 1 since no flow can be sent between the two nodes. Since the objective is to minimize the weighted sum of y_C, the variable y_C will be set to 0 unless the logical topology is disconnected. Therefore, the variable y_C represents whether C is a cross-layer cut. This is true even if the binary constraint on y_C is relaxed.

2.4.2 Approximate Formulations

Although MCLC_MAX gives us an exact formulation to maximize MCLC, the formulation may have a large number of variables and constraints, and is therefore infeasible to solve in practice, even if all the integer variables are relaxed. Therefore, for the rest of the section, we consider approximate formulations whose objective values are *lower bounds* to the MCLC. These formulations are much simpler than MCLC_MAX. This makes it possible to develop survivable lightpath routing algorithms based on these simpler formulations. In particular, in Section 2.4.3 we discuss how to use randomized rounding [90] based on these formulations as a heuristic to approximate MCLC maximization. Note that since MCLC is $O(\log n)$ inapproximable, polynomial time algorithms with approximation guarantees within this factor are unlikely to exist. Therefore, we will instead evaluate the performance of our algorithms via simulation in Section 2.5.

All of the formulations introduced in this section are based on multi-commodity flows, where each lightpath is considered a commodity to be routed over the physical network. Given the physical network $G_P = (V_P, E_P)$ and the logical network $G_L = (V_L, E_L)$, the multi-commodity flow for a lightpath routing can be generally formulated as follows:

$$\mathsf{MCF}_{\mathcal{X}}: \quad \text{Minimize } \mathcal{X}(f), \quad \text{subject to:}$$

$$f_{ij}^{st} \in \{0,1\}$$

$$\{f_{ij}^{st} : (i,j) \in E_P\} \text{ forms an } (s,t)\text{-path}, \ \forall (s,t) \in E_L, \tag{2.7}$$

where f is the variable set that represents the lightpath routing, such that $f_{ij}^{st} = 1$ if and only if lightpath (s,t) uses physical fiber (i,j) in its route; and the objective $\mathcal{X}(f)$ is a function of the lightpath routing f that captures the survivability of the layered network.

For WDM networks where the wavelength continuity constraint is present [29,110], the above formulation can be extended to capture the wavelength assignment aspect. In that case, the wavelength assignment can be modelled by replacing the variable set f_{ij}^{st} by $f_{ij\lambda}^{st}$, which equals 1 if and only if lightpath (s,t) uses wavelength λ on physical link (i,j). Constraint (2.7) can be easily extended to restrict that, for each logical link (s,t), $\{f_{ij\lambda}^{st} = 1\}$ forms an (s,t) physical path along one of the wavelengths. To make sure that any wavelength λ on a physical fiber is used by at most one lightpath, the following constraint will be added:

$$\sum_{(s,t)\in E_L} f_{ij\lambda}^{st} \leq 1 \quad \forall (i,j) \in E_P, \forall \lambda. \tag{2.8}$$

Similar formulations based on multi-commodity flows with wavelength continuity constraint have been proposed to solve the Routing and Wavelength Assignment (RWA) problem of WDM networks [14,85], where the objective is to minimize the number of lightpaths that traverse the same fiber. The key difference in the problem studied in this chapter is in the objective function $\mathcal{X}$, which should instead describe the survivability of the lightpath routing. To focus on the survivability aspect of the problem, the wavelength continuity constraint will be omitted in the formulations below. However, in cases where the wavelength continuity constraint is necessary, all these formulations can be extended as discussed above.

Simple Multi-Commodity Flow Formulations

Ideally, to ensure that the lightpath routing is survivable against the largest number of failures, the objective function $\mathcal{X}(f)$ should express the MCLC value of the lightpath routing given by f. However, since simple formulations to maximize the MCLC directly are difficult to find, we use an objective that approximates the MCLC value.

In our formulation, each lightpath is assigned a weight w. The objective function ρ_w measures the maximum *load* of the fibers, where the *load* is defined to be the total lightpath weight carried by the fiber. The intuition is that the multi-commodity flow formulation will try to spread the weight of the lightpaths across multiple fibers, thereby minimizing the impact of any single fiber failure.

We can formulate an integer linear program with such an objective as follows:

$$\begin{aligned}
\mathsf{MCF}_w : \quad & \text{Minimize } \rho_w, \quad \text{subject to:} \\
& \rho_w \geq \sum_{(s,t)\in E_L} w(s,t) f_{ij}^{st} \quad \forall (i,j) \in E_P \\
& f_{ij}^{st} \in \{0,1\} \\
& \{f_{ij}^{st} : (i,j) \in E_P\} \text{ forms an } (s,t)\text{-path}, \ \forall (s,t) \in E_L
\end{aligned}$$

As we will prove in Theorem 2.11, with a careful choice of the weight function w, the value $\frac{1}{\rho_w}$ gives a lower bound on the MCLC. Therefore, a lightpath routing with a low ρ_w value is guaranteed to have a high MCLC.

The routing strategy of the algorithm is determined by the weight function w. For example, if w is set to 1 for all lightpaths, the integer program will minimize the number of lightpaths traversing the same fiber. Effectively, this will minimize the number of disconnected lightpaths in the case of a single fiber failure.

In order to customize MCF_w towards maximizing the MCLC of the solution, we propose a different weight function w_{MinCut} that captures the connectivity structure of the logical topology. For each edge $(s,t) \in E_L$, we define $w_{MinCut}(s,t)$ to be $\frac{1}{|MinCut_L(s,t)|}$, where $MinCut_L(s,t)$ is the minimum (s,t)-cut in the logical topology. Therefore, if an edge (s,t) belongs to a smaller cut, it will be assigned a higher weight. The algorithm will therefore try to avoid putting these small cut edges on the same fiber.

If w_{MinCut} is used as the weight function used in MCF_w, we can prove the following relationship between the objective value ρ_w of a feasible solution to MCF_w and the Weighted Load Factor of the associated lightpath routing:

Theorem 2.11 *For any feasible solution f of* $\mathsf{MCF_w}$ *with w_{MinCut} as the weight function, $\frac{1}{\rho_w} \leq WLF$.*

Proof. By definition of the weight function w_{MinCut}, given any $S \subset V_L$, every edge in $\delta(S)$ has weight at least $\frac{1}{|\delta(S)|}$. Therefore, we have:

$$\sum_{(s,t)\in\delta(S)} w(s,t) \geq \sum_{(s,t)\in\delta(S)} \frac{1}{|\delta(S)|} = 1 \tag{2.9}$$

Now consider the lightpath routing associated with f. For any logical cut $\delta(S)$, the maximum fraction of weight inside the cut carried by a fiber is:

$$\begin{aligned}
&\max_{(i,j)\in E_P} \frac{\sum\limits_{(s,t)\in\delta(S)} w(s,t) f_{ij}^{st}}{\sum\limits_{(s,t)\in\delta(S)} w(s,t)} \\
&\leq \max_{(i,j)\in E_P} \sum_{(s,t)\in\delta(S)} w(s,t) f_{ij}^{st}, \qquad \text{by Equation (2.9)} \\
&\leq \max_{(i,j)\in E_P} \sum_{(s,t)\in E_L} w(s,t) f_{ij}^{st} \\
&\leq \rho_w.
\end{aligned}$$

In other words, no fiber in the network is carrying more than a fraction ρ_w of the weight in any cut. This gives us a feasible solution to the Weighted Load Factor formulation $\mathsf{M_{WLF}}$, where each variable w_{st} is assigned the value of $w_{MinCut}(s,t)$, and the variable z is assigned the value of ρ_w. As a result, the Weighted Load Factor, defined to be the maximum value of $\frac{1}{z}$ among all feasible solutions to $\mathsf{M_{WLF}}$, must be at least $\frac{1}{\rho_w}$. □

As a result of Theorems 2.10 and 2.11, the MCLC of a lightpath routing is lower bounded by the value of $\frac{1}{\rho_w}$, which the algorithm will try to maximize.

Enhanced Multi-Commodity Flow Formulation

As we have discussed in Section 2.3.2, the Weighted Load Factor provides a good lower bound on the MCLC of a lightpath routing. Here we propose another multi-commodity flow based formulation whose objective function approximates the Weighted Load Factor of a lightpath routing. The formulation, denoted as $\mathsf{MCF_{LF}}$, can be written as follows:

$$\begin{aligned}
&\mathsf{MCF_{LF}}: \quad \text{Minimize } \gamma, \quad \text{subject to:}\\
&\gamma|\delta(S)| \geq \sum_{(s,t)\in\delta(S)} f_{ij}^{st} \quad \forall (i,j) \in E_P, S \subset V_L\\
&\qquad f_{ij}^{st} \in \{0,1\}\\
&\qquad \{f_{ij}^{st} : (i,j) \in E_P\} \text{ forms an } (s,t)\text{-path}, \ \ \forall (s,t) \in E_L
\end{aligned}$$

Essentially, the formulation optimizes the *unweigthed* Load Factor of the lightpath routing, (i.e., all weights equal one), by minimizing the maximum fraction of a logical cut carried by a single fiber. As this formulation provides a constraint for each logical cut, it captures the impact of a single fiber cut on the logical topology in much greater detail. The following theorem shows that for any lightpath routing, its associated Load Factor value $\frac{1}{\gamma}$ gives a tighter lower bound than $\frac{1}{\rho_w}$, given by the MCF_w formulation.

Theorem 2.12 *For any lightpath routing, let ρ_w be its associated objective value in the formulation MCF_w with w_{MinCut} as the weight function, and let γ be its associated objective value in the formulation MCF_{LF}. In addition, let WLF be its Weighted Load Factor. Then:*

$$\frac{1}{\rho_w} \leq \frac{1}{\gamma} \leq WLF.$$

Proof. The value $\frac{1}{\gamma}$ is the objective value for the formulation $\mathsf{M_{WLF}}$ in Section 2.3.2 when all logical links have weight 1. This gives a feasible solution to $\mathsf{M_{WLF}}$, and implies that $WLF \geq \frac{1}{\gamma}$.

To prove that $\frac{1}{\rho_w} \leq \frac{1}{\gamma}$, we consider the physical link (i,j) and logical cut set $\delta(S)$ where (i,j) carries a fraction γ of the logical links in $\delta(S)$. Let L_{ij} be the set of logical links in E_L carried by (i,j). Therefore, we have $\gamma = \frac{|L_{ij}\cap\delta(S)|}{|\delta(S)|}$. In addition, by the definition of ρ_w, we have

$$\begin{aligned}\rho_w &\geq \sum_{(s,t)\in L_{ij}} w(s,t)\\ &\geq \sum_{(s,t)\in L_{ij}\cap\delta(S)} w(s,t)\\ &\geq \sum_{(s,t)\in L_{ij}\cap\delta(S)} \frac{1}{|\delta(S)|}\\ &= \frac{|L_{ij}\cap\delta(S)|}{|\delta(S)|} = \gamma.\end{aligned}$$

This implies $\frac{1}{\rho_w} \leq \frac{1}{\gamma}$. □

Therefore, the formulation $\mathsf{MCF}_{\mathsf{LF}}$ gives a lightpath routing that is optimized for a better lower bound on the MCLC. However, this comes at the cost of a larger number of constraints and solving such an integer program may not be feasible in practice. Therefore, we next introduce a randomized rounding technique that approximates the optimal lightpath routing by solving the linear relaxation of the integer program. As we will see in Section 2.5, the randomized rounding technique significantly speeds up the running time of the algorithm without observable degradation in the MCLC performance. This offers a practical alternative to solving the integer program formulations introduced in this section.

2.4.3 Randomized Rounding for Lightpath Routing

While the multi-commodity flow integer program formulations discussed in the previous section introduce a novel way to route lightpaths in a survivable manner, such an approach may not scale to large networks, due to the inherent complexity of solving integer programs. In order to circumvent the computational difficulty, we apply the *randomized rounding* technique, which is able to quickly obtain a near-optimal solution to the integer program. Randomized rounding has previously been used to solve multi-commodity flow problems to minimize the link load [14, 90], and its performance guarantee is studied in [90].

Given any multi-commodity flow based integer formulation, the following algorithm $\mathsf{RANDOM_k}$ describes the randomized rounding algorithm that computes a lightpath routing based on the formulation.

Algorithm 1 $\mathsf{RANDOM_k}$

```
1: Compute the optimal fractional solution $f$ to the linear relaxation of the multi-commodity flow integer program. For each lightpath $(s, t)$, the values of $f_{ij}^{st}$ represent a flow from $s$ to $t$ with a total flow value of 1.
2: For each lightpath $(s, t)$, decompose the solution $f_{ij}^{st}$ into flow paths, each with weight equal to the flow value of the path.
3: for $i = 1, 2, \ldots, k$, do:
       Create a random lightpath routing $R_i$: For each lightpath $(s, t)$, randomly pick one path from the set of flow paths generated in Step 2, using the path weights as the probabilities.
4: Return the $R_i$ with the highest Min Cross Layer Cut value.
```

The parameter k specifies the number of trials in the process of random lightpath routing generation. The higher the value of k, the more likely the algorithm will encounter a lightpath routing with a high MCLC value.

Although the last step requires the MCLC computation of the lightpath routings, the integer program $\mathsf{M_{MCLC}}$ contains only $|E_P|$ binary variables, which is much fewer than the $|E_P||E_L|$ variables contained in the multi-commodity flow formulations. Therefore, the randomized algorithm runs considerably faster than the integer program algorithm. In the next section, we will compare the performance of the two algorithms, both in terms of running time and quality of the solution.

2.5 Simulation

In this section, we discuss our simulation results for the algorithms introduced in Section 2.4. We first compare the lightpath routing algorithms by solving the ILP directly and by randomized rounding. Next, we compare the survivability performance among different formulations. Finally, we investigate the different lower bounds of MCLC, and their effects on the MCLC value of the lightpath routing when used as an optimization objective.

ILP vs Randomized Rounding

In this experiment, we use the NSFNET (Figure 2-3) as the physical topology. The network is augmented to have connectivity 4, which makes it possible to study the performance of the algorithms where a higher MCLC value is possible. We generated 350 random logical topologies with connectivity at least 4, and size ranging from 6 to 12 nodes. Using the formulation MCF_w with weight function $w_{MinCut}(s,t)$ introduced in Section 2.4.2 as our benchmark, we compare the performance of RANDOM_{10} against solving the ILP optimally.

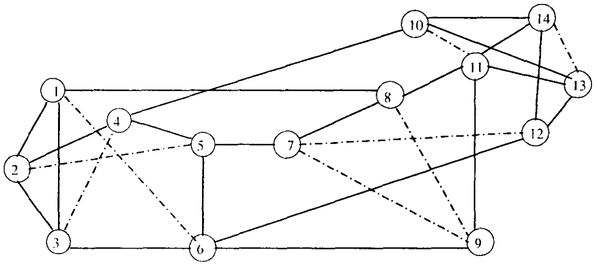


Figure 2-3: The augmented NSFNET. The dashed lines are the new links.

Table 2.1 compares the average running time between the algorithms ILP and RANDOM_{10} on various logical topology size. All simulations are run on a Xeon E5420 2.5GHz workstation with 4GB of memory, using CPLEX to solve the integer and linear programs. As the number of logical nodes increases, the running time for the integer program algorithm ILP increases tremendously. On the other hand, there is no observable growth in the average running time for the algorithm RANDOM_{10}, which is less than a minute. In fact, our simulation on larger networks shows that the algorithm ILP often fails to terminate within a day when the network size goes beyond 12 nodes. On the other hand, the algorithm RANDOM_{10} for MCF_w is able to terminate consistently within 2 hours for very large instances with a 100-node physical topology and 50-node logical topology. This shows that the randomized approach is a much more scalable solution to compute survivable lightpath routings.

In Figure 2-4, the survivability performance of the randomized algorithm is compared with its ILP counterpart. Each data point in the figure is the MCLC average of 50 random instances with the given logical network size. As our result shows, the

Logical Topology Size	Average Running Time (seconds)	
	ILP	RANDOM_{10}
6	33.2	31.9
7	50.5	33.9
8	660.0	30.1
9	1539.0	26.4
10	3090.6	32.3
11	8474.5	32.0
12	15369.7	29.7

Table 2.1: Average running time of ILP and RANDOM_{10}.

lightpath routings produced by RANDOM_{10} have higher MCLC values than solving the ILP optimally. This is because the objective value for ILP MCF_w is a *lower bound* on MCLC. As we will see in Section 2.5, this lower bound is often not tight enough to accurately reflect the MCLC value, which means that the optimal solution to the ILP does not necessarily yield a lightpath routing with maximum MCLC. On the other hand, the randomized algorithm generates lightpath routings non-deterministically based on the optimal fractional solution of MCF_w. Therefore, it approximates the lightpath routing given by the ILP, with an additional randomization component to explore better solutions. When the randomized rounding process is repeated many times, the algorithm often encounters a solution that is even better than the one given by the ILP.

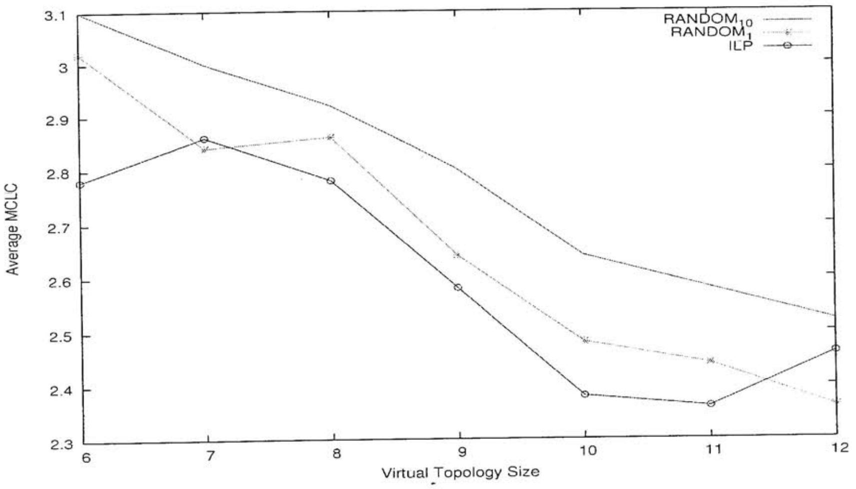


Figure 2-4: MCLC performance of randomized rounding vs ILP.

To sum up, randomized rounding provides an efficient alternative to solving integer programs without observable quality degradation. This allows us to experiment with more complex formulations in larger networks where solving the integer programs optimally is infeasible. In the next section, we will compare the different formulations introduced in Section 2.4.2, using randomized rounding to compute the lightpath routings.

Lightpath Routing with Different Formulations

In this experiment, we study the survivability performance of the lightpath routings generated by the formulations introduced in Section 2.4. We use the 24-node USIP network (Figure 2-5), augmented to have connectivity 4, as the physical topology. We generate 500 random graphs with connectivity 4 and size ranging from 6 to 15 nodes as logical topologies.

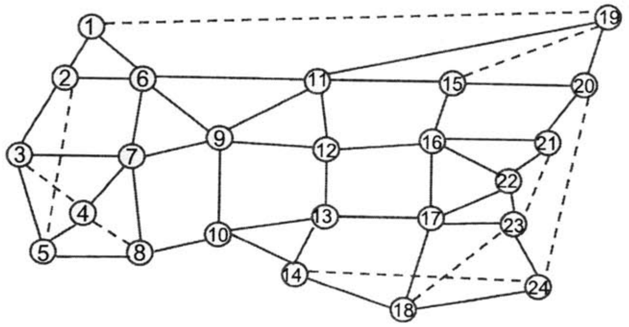


Figure 2-5: The augmented USIP network. The dashed lines are the new links.

We compare the MCLC performance of the lightpath routings generated by the randomized rounding algorithm, RANDOM_{100}, on the following formulations:

1. Multi-Commodity Flow MCF_w, using identity function as the weight function, i.e., $w(s,t) = 1$ for all $(s,t) \in E_L$ (Identity);

2. Multi-Commodity Flow MCF_w, using the weight function w_{MinCut}; introduced in Section 2.4.2 (MinCut);

3. Enhanced Multi-Commodity Flow $\mathsf{MCF}_{\mathsf{LF}}$ (LF).

For comparison, we also run randomized rounding on the Survivable Lightpath Routing formulation (SURVIVE), introduced in [76], which computes the lightpath routing that minimizes the total fiber hops, subject to the constraint that the MCLC must be at least two.

Figure 2-6 compares the average MCLC values of the lightpath routings computed by the four different algorithms. Overall, the formulations introduced in this chapter achieve better survivability than SURVIVE. This is because these formulations try to maximize the MCLC in their objective functions, whereas SURVIVE minimizes the physical hops. Therefore, even though SURVIVE does well in finding a survivable routing (i.e. MCLC$\geq$2), the new formulations are able to achieve even higher MCLC values, which allow more physical failures to be tolerated.

To further verify the survivability performance of the lightpath routings from a different perspective, for each lightpath routing, we simulated the scenario where each physical link fails independently with probability 0.01. Figure 2-7 shows the average probability that the logical topology becomes disconnected under this scenario. The result is consistent with Figure 2-6, as lightpaths routings with higher MCLC values can tolerate more physical failures, and the logical topologies are thus more likely to stay connected.

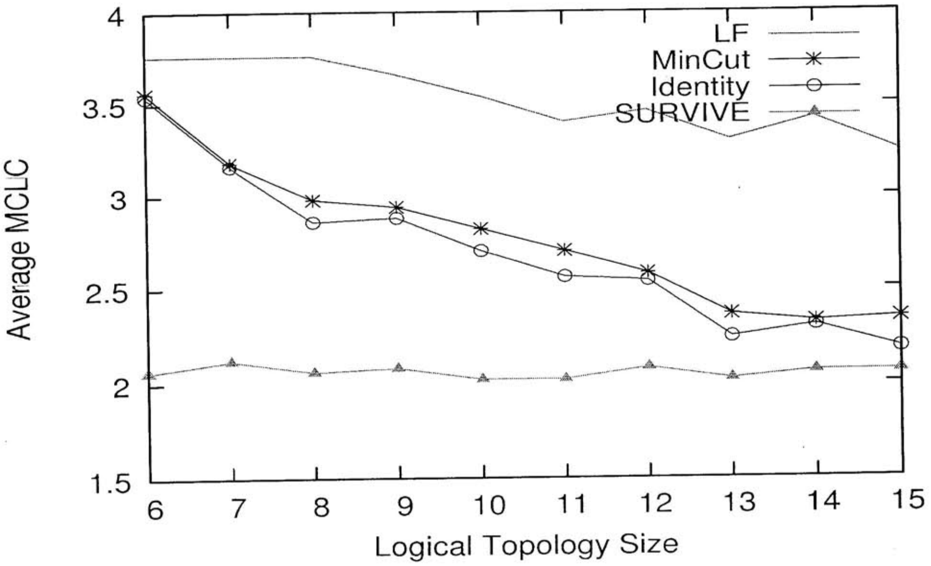


Figure 2-6: MCLC performance of different lightpath routing formulations.

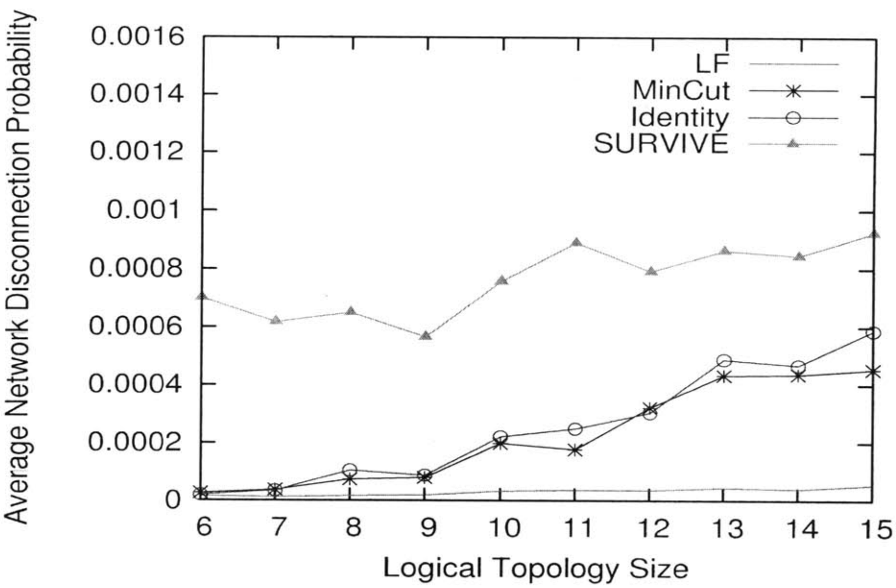


Figure 2-7: Probabilty that logical topology becomes disconnected if physical links fail independently with probability 0.01.

The quality of the lightpath routing also depends on the graph structures captured by the formulations. Compared with $\mathsf{MCF}_{\mathsf{Identity}}$, the formulation $\mathsf{MCF}_{\mathsf{MinCut}}$ uses a weight function that captures the connectivity structure of the logical topology. As a result, the algorithm will try to avoid putting edges that belong to smaller cuts onto the same physical link, thereby minimizing the impact of a physical link failure on these critical edges. This allows the algorithm $\mathsf{MCF}_{\mathsf{MinCut}}$ to produce lightpath routings with higher MCLC values than $\mathsf{MCF}_{\mathsf{Identity}}$.

The enhanced formulation $\mathsf{MCF}_{\mathsf{LF}}$ captures the connectivity structure of the logical topology in much greater detail, by having a constraint to describe the impact of a physical link failure to each logic cut. As a result, the algorithm based on this formulation is able to provide lightpath routings with the highest MCLC values.

Lower Bound Comparison

In Theorem 2.12 we establish different lower bounds for the MCLC. In this experiment, we measure these lower bound values for 500 different lightpath routings, and compare them to the actual MCLC values.

As Figure 2-8 shows, the Weighted Load Factor is a very close approximation of the Min Cross Layer Cut. Among the 500 routings being investigated, the two metrics are identical in 368 cases. This suggests a tight connection between the two metrics, which also justifies the choice of such metrics as survivability measures.

The figure also reveals a strong correlation between the MCLC performance and the tightness of the lower bounds given by the multi-commodity flow formulations in Section 2.4.2. Compared to MCF_w, the formulation $\mathsf{MCF}_{\mathsf{LF}}$ provides an objective value that is closer to the actual MCLC value of the lightpath routing. This translates to better lightpath routings, as we saw in Figure 2-6. Since there is still a large gap between the $\mathsf{MCF}_{\mathsf{LF}}$ objective value and the MCLC value, this suggests room for further improvement with a formulation that gives a better MCLC lower bound.

To summarize this section, a good formulation that properly captures the cross-layer connectivity structure is essential for generating lightpath routings with high survivability. Combined with randomized rounding, it gives a powerful tool for designing highly survivable layered networks.

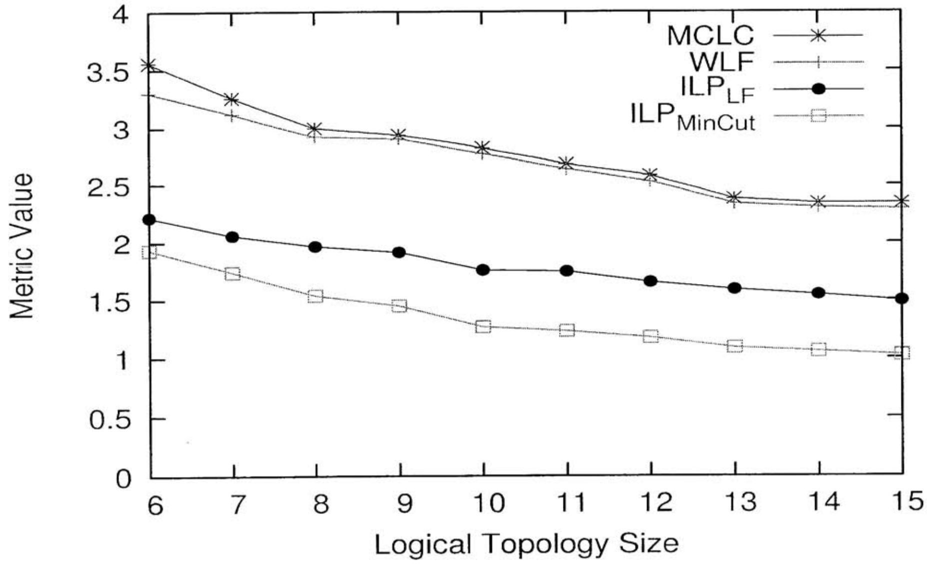


Figure 2-8: Comparison among Min Cross Layer Cut (MCLC), Weighted Load Factor (WLF) , and the optimal values of $\mathsf{ILP}_{\mathsf{LF}}$ and $\mathsf{ILP}_{\mathsf{MinCut}}$.

2.6 Conclusion

In this chapter, we introduce the problem of maximizing the connectivity of layered networks. We show that survivability metrics in multi-layer networks have significantly different meaning than their single-layer counterparts. We propose two survivability metrics, the Min Cross Layer Cut and the Weighted Load Factor, that measure the connectivity of a multi-layer network, and develop linear and integer formulations to compute these metrics. In addition, we use the metric Min Cross Layer Cut as the objective for the survivable lightpath routing problem, and develop multi-commodity flow formulations to approximate this objective. We show, through simulations, that our algorithms produce lightpath routings with significantly better Min Cross Layer Cut values than existing survivable lightpath routing algorithms.

Our simulations show that a good formulation, combined with the randomized rounding technique, provides a powerful tool for generating highly survivable layered networks. Therefore, an important direction for future research is to establish a better formulation for the lightpath routing problem that maximizes the Min Cross Layer Cut. The multi-commodity flow formulation introduced in this chapter approximates the Min Cross Layer Cut by using its lower bound as the objective function. However, this lower bound is often not very close to the actual Min Cross Layer Cut value. A better objective function, such as the Weighted Load Factor, would significantly improve the proposed lightpath routing algorithms.

The similarity between the Min Cross Layer Cut and the Weighted Load Factor is also intriguing. Our simulation results demonstrated a very tight connection between the two metrics. This observation might reflect certain property of cross-layer network connectivity that are yet to be discovered and formalized. A better understanding of how these metrics relate to each other will possibly lead to important insights into the cross-layer survivability problem.

2.7 Chapter Appendix

2.7.1 Proof of Theorem 2.1

Theorem 2.1: Let G_L be a logical topology with two nodes s and t, connected by n lightpaths $E_L = \{e_1, e_2, \dots, e_n\}$, and let $\mathcal{R} = \{R_1, R_2, \dots, R_k\}$ be a family of subsets of E_L where each $|R_i| \geq 2$. There exists a physical topology $G_P = (V_P, E_P)$ and lightpath routing of G_L over G_P, such that:

1. there are exactly k fibers in E_P, denoted by $F = \{f_1, f_2, \dots, f_k\}$, that are used by multiple lightpaths;

2. for each fiber $f_i \in F$, the set of lightpaths using the fiber f_i is R_i.

Proof. Given a logical topology $G_L = (V_L, E_L)$ with two nodes s and t connected by n lightpaths $E_L = \{e_1, e_2, \dots, e_n\}$, and $\mathcal{R} = \{R_1, R_2, \dots, R_k\}$ be the family of subsets of E_L, we construct a physical topology and lightpath routing that satisfy the conditions specified in the theorem.

- **Physical Topology**:

  The physical topology contains the two end nodes s and t in the logical network. In addition, between the two end nodes, there are n groups of nodes. Each group i containing $k + 1$ nodes, namely $x_0^i, x_1^i, \dots, x_k^i$. For any $i \in \{1, \dots, n\}, j \in \{1, \dots k\}$, there is an edge connecting nodes x_{j-1}^i and x_j^i. In addition, s is connected to x_0^i and x_k^i is connected to t for all $i \in \{1, \dots, n\}$. In other words, in the physical network we have constructed so far, there are n edge disjoint paths connecting s and t, and each path has $k + 2$ edges.

  Next, we add k pairs of nodes $\{(y_1, z_1), \dots, (y_k, z_k)\}$ to the physical network, where each node pair (y_j, z_j) is connected by an edge. Finally, we connect x_{j-1}^i to y_j and z_j to x_j^i for all $i \in \{1, \dots, n\}, j \in \{1, \dots, k\}$.

- **Lightpath Routing**:

  We will define a route in the physical topology for each lightpath e_i. Each route

l_i will contain $k+2$ segments:

$$s \rightsquigarrow x_0^i \rightsquigarrow x_1^i \rightsquigarrow \ldots \rightsquigarrow x_k^i \rightsquigarrow t.$$

Segments $s \rightsquigarrow x_0^i$ and $x_k^i \rightsquigarrow t$ will take the direct edges $s \rightarrow x_0^i$ and $x_k^i \rightarrow t$ respectively as their routes. The routes for other segments depend on whether e_i is in R_j:

- If $e_i \in R_j$, the route for $x_{j-1}^i \rightsquigarrow x_j^i$ is $x_{j-1}^i \rightarrow y_j \rightarrow z_j \rightarrow x_j^i$;
- If $e_i \notin R_j$, the route for $x_{j-1}^i \rightsquigarrow x_j^i$ is $x_{j-1}^i \rightarrow x_j^i$.

Figure 2-9 shows the physical topology and lightpath routing constructed from a two-node logical topology with $\mathcal{R} = \{\{1,2\}, \{2\}, \{1,3\}, \{1\}\}$.

By construction, all fibers except $\{(y_1, z_1), \ldots, (y_k, z_k)\}$ are used by at most one lightpath. Also, a lightpath e_j uses fiber (y_i, z_i) if and only if e_j is in R_i. In other words, there are exactly k fibers, $(y_1, z_1), \ldots, (y_k, z_k)$, that are used by multiple lightpaths, and each fiber (y_i, z_i) is used by the lightpaths in R_i. □

2.7.2 Proof of Theorem 2.2

Let $\mathsf{MinSPS_{st}}$ be the size of the minimum survivable path set between the logical nodes s and t. Theorem 2.2 describes the relationship between the value of $\mathsf{MinSPS_{st}}$ and the relaxed Max Flow, $\mathsf{MaxFlow^R_{st}}$, between the two nodes:

Theorem 2.2: $\mathsf{MinSPS_{st}} \leq \left\lfloor \frac{\log |E_P|}{\log \mathsf{MaxFlow^R_{st}}} \right\rfloor + 1$.

Proof. Let $\mathcal{P}_{st}$ and E_P be the set of logical $s-t$ paths and the set of physical links respectively. For each $s-t$ path $p \in \mathcal{P}_{st}$, denote the set of physical links used by p as $L(p)$. We first construct a bipartite graph on the node set $(\mathcal{P}_{st}, E_P)$. There is an edge $(p, l) \in \mathcal{P}_{st} \times E_P$ if and only if the $s-t$ path p does *not* use physical link l, i.e., $l \notin L(p)$. In other words, the edge (p, l) is in the bipartite graph if and only if the path p survives the failure of physical link l.

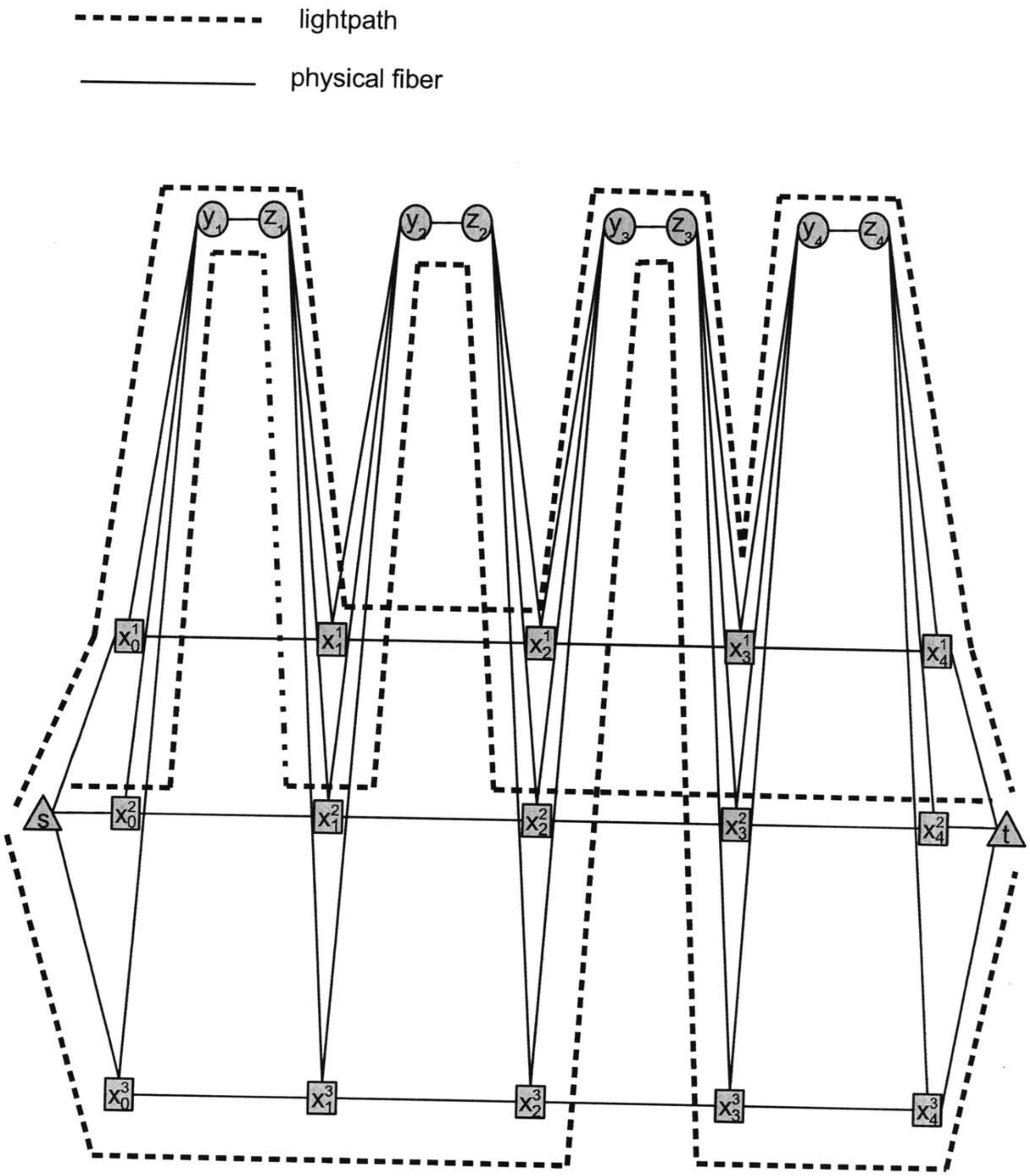


Figure 2-9: The physical topology and lightpath routing on three lightpaths between two logical nodes s and t, and lightpath-sharing relationship $\mathcal{R} = \{\{1,2\},\{2\},\{1,3\},\{1\}\}$.

We prove the theorem by explicitly constructing a survivable path set with size at most $\left\lfloor \frac{\log |E_P|}{\log \mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}}} \right\rfloor + 1$, using the bipartite graph. Algorithm $\mathsf{SPS}_{\mathsf{GREEDY}}$ describes a greedy algorithm that constructs the path set by repeatedly selecting $s-t$ paths and removing physical links whose failures the selected path can survive. When the algorithm terminates, every physical link failure is survived by a selected path in the output. Therefore, the algorithm gives a survivable path set.

Algorithm 2 $\mathsf{SPS}_{\mathsf{GREEDY}}$

```
1: $P := \emptyset$, $S := E_P$
2: while $S \neq \emptyset$: do:
       - Select $p \in \mathcal{P}_{st}$ with the largest node degree in the bipartite graph.
       - $P := P \cup \{p\}$, $S := S \backslash L(p)$
       - Remove nodes $p$ and $L(p)$ from the bipartite graph.
3: Return $P$
```

The key observation for this algorithm is that, every iteration of the algorithm removes a constant fraction of remaining nodes in E_P. We state this result as the following lemma:

Lemma 2.13 *Let B^i be the bipartite graph at the beginning of the i^{th} iteration of the algorithm, where the remaining node sets for E_P and $\mathcal{P}_{st}$ are E^i_P and $\mathcal{P}^i_{st}$ respectively. There exists a node in $\mathcal{P}^i_{st}$ with node degree at least $\frac{|E^i_P|(\alpha-1)}{\alpha}$, where α is the optimal value for the formulation* $\mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}}$.

Proof. Suppose $\{f^*_p | p \in \mathcal{P}_{st}\}$ is the optimal solution for $\mathsf{MaxFlow}^{\mathsf{R}}_{\mathsf{st}}$, such that:

$$\sum_{p \in \mathcal{P}_{st}} f^*_p = \alpha.$$

For the purpose of analysis, for each edge $(p, l) \in \mathcal{P}^i_{st} \times E^i_P$ in the bipartite graph, we assign the edge a weight f^*_p.

For each node v in the bipartite graph, let $d(v)$ be its node degree, and we define its weight $w(v)$ to be sum of the weight of its incident edges. Then we have:

$$\sum_{p \in \mathcal{P}^i_{st}} \frac{w(p)}{d(p)} = \sum_{p \in \mathcal{P}^i_{st}} f^*_p \leq \alpha. \tag{2.10}$$

For each node l in E_P^i, its neighbors in $\mathcal{P}_{st}^i$ are the same as its neighbors in $\mathcal{P}_{st}$, since otherwise it should have already been removed from the bipartite graph. Its node weight is:

$$\begin{aligned} w(l) = &\sum_{p \in \mathcal{P}_{st}^i : l \notin L(p)} f_p^* = \sum_{p \in \mathcal{P}_{st} : l \notin L(p)} f_p^* \\ = &\sum_{p \in \mathcal{P}_{st}} f_p^* - \sum_{p \in \mathcal{P}_{st} : l \in L(p)} f_p^* \\ \geq &\ \alpha - 1, \qquad \text{since } \sum_{p : l \in L(p)} f_p^* \leq 1, \text{ by Equation (2.1).} \end{aligned}$$

Therefore the total weight for the nodes in E_P^i is at least $|E_P^i|(\alpha - 1)$, which implies:

$$\sum_{p \in \mathcal{P}_{st}^i} w(p) \geq |E_P^i|(\alpha - 1). \tag{2.11}$$

Let d_{max} be the largest node degree among the nodes in $\mathcal{P}_{st}^i$. We have:

$$d_{max} \geq \frac{\sum_{p \in \mathcal{P}_{st}^i} w(p)}{\sum_{p \in \mathcal{P}_{st}^i} \frac{w(p)}{d(p)}} \geq \frac{|E_P^i|(\alpha - 1)}{\alpha}, \text{by Equations (2.10) and (2.11).}$$

Therefore, the set $\mathcal{P}_{st}^i$ contains a node with degree at least $\frac{|E_P^i|(\alpha-1)}{\alpha}$. □

As a result of Lemma 2.13, every iteration of the algorithm removes a fraction of $\frac{\alpha-1}{\alpha}$ nodes of E_P^i from the bipartite graph. Therefore, after the i^{th} path is selected, the number of nodes in E_P that remain in the bipartite graph is at most $(1 - \frac{\alpha-1}{\alpha})^i |E_P|$. The algorithm will terminate as soon as:

$$(1 - \frac{\alpha - 1}{\alpha})^i |E_P| < 1, \text{which implies } i > \frac{\log |E_P|}{\log \alpha}.$$

Therefore, the algorithm returns a survivable path set with size $\lfloor \log_\alpha |E_P| \rfloor + 1$. □

2.7.3 Proof of Theorem 2.7

Theorem 2.7: Given the lightpath routing for a multi-layer network $\mathcal{G} = (G_P, G_L)$, finding its Minimum Cross-Layer Spanning Tree is NP-hard.

Proof. We prove the theorem by constructing a reduction from the NP-Hard *Minimum Label Spanning Tree* problem [26]:

Minimum Label Spanning Tree: *Given a graph $G = (V, E)$, and a set of labels $\mathcal{L} = \{L_1, \dots, L_m\}$. Each edge $e \in E$ is associated with a set of labels $\mathcal{L}_e \subseteq \mathcal{L}$. Find a spanning tree T of G with minimum number of labels, that is, the value $| \cup_{e \in T} \mathcal{L}_e|$ is minimized.*

Given an instance of the Minimum Label Spanning Tree problem, we will construct an instance of the Minimum Cross-Layer Spanning Tree problem, which consists the the physical topology $G_P = (V_P, E_p)$, logical topology $G_L = (V_L, E_L)$ and lightpath routing.

Logical Topology: The logical topology G_L is the same as the graph G in the Minimumm Label Spanning Tree problem.

Physical Topology: The physical topology contains all the nodes in the logical topology. In addition, for each label $L_i \in \mathcal{L}$, we add a pair of nodes p_i and q_i, with a physical link (p_i, q_i) connecting the two nodes.

Next, for each logical link $(s, t) \in E_L$, we denote $h_0^{st} = s$ and $h^{st}_{|\mathcal{L}|} = t$. Between h_0^{st} and $h^{st}_{|\mathcal{L}|}$, we insert a sequence of $2 * |\mathcal{L}| - 1$ physical nodes $\left\{x_1^{st}, h_1^{st}, \dots, h^{st}_{|\mathcal{L}|-1}, x^{st}_{|\mathcal{L}|}\right\}$, and construct a physical path between the two nodes: $h_0^{st} \rightarrow x_1^{st} \rightarrow h_1^{st} \dots \rightarrow x^{st}_{|\mathcal{L}|} \rightarrow h^{st}_{|\mathcal{L}|}$.

Finally, for each label $L_i \in \mathcal{L}, (s, t) \in G$ and logical link $(s, t) \in E_L$, we add two physical links $(h^{st}_{i-1}, p_i), (q_i, h^{st}_i)$.

Lightpath Routing: For each logical link (s, t), the lightpath routing for (s, t) consists of $|\mathcal{L}|$ segments $s \rightsquigarrow h_1^{st} \rightsquigarrow h_2^{st} \dots \rightsquigarrow h^{st}_{|\mathcal{L}|-1} \rightsquigarrow t$.

For each $i \in \{1, \dots, |\mathcal{L}|\}$, the route for each segment $h^{st}_{i-1} \rightsquigarrow h^{st}_i$ depends on whether the edge (s, t) has label L_i in the original Minimum Label Spanning Tree instance. If the edge has label L_i, then the segment $h_{i-1}^{st} \rightsquigarrow h_i^{st}$ takes on the route $h_{i-1}^{st} \to p_i \to q_i \to h_i^{st}$. Otherwise, $h_{i-1}^{st} \rightsquigarrow h_i^{st}$ takes on the route $h_{i-1}^{st} \to x_i^{st} \to h_i^{st}$.

Under this lightpath routing, only physical links of the form (p_i, q_i) can be shared by mutilple logical links. Other physical links can be used by at most one logical link. We call the first kind of physical links *non-exclusive* physical links, and the others *exclusive* physical links.

Therefore, each segment $h_{i-1}^{st} \rightsquigarrow h_i^{st}$ traverses exactly two exclusive physical links, and in addition one non-exclusive link if the edge (s, t) has label L_i in the corresponding Minimum Label Spanning Tree problem. In other words, each logical link (s, t) traverses $2|\mathcal{L}|$ exclusive physical links and $|\mathcal{L}_{st}|$ non-exclusive physical links, where $\mathcal{L}_{st}$ is the set of labels associated with (s, t).

An example of the reduction is shown in Figures 2-10 and 2-11.

We prove the following lemma, which implies that finding the minimum label spanning tree reduces to finding the minimum cross-layer spanning tree.

Lemma 2.14 *Let α be the number of labels associated with the optimal solution for the Minimum Label Spanning Tree instance, and let β be the number of physical links in Minimum Cross-Layer Spanning Tree instance under the reduction. Then $\beta = 2(n-1)|\mathcal{L}| + \alpha$.*

Proof. Since at least $n - 1$ logical links must survive if a cross-layer spanning tree survives, and each logical links uses exactly $2|\mathcal{L}|$ exclusive fibers, every cross-layer spanning tree contains at least $2(n-1)|\mathcal{L}|$ exclusive fibers.

First, suppose T is the minimum label spanning tree in the Minimum Label Spanning Tree problem with α labels. In the corresponding Minimum Cross-Layer Spanning Tree problem, T is also a spanning tree for the logical topology where each logical link $(s, t) \in T$ traverses $2|\mathcal{L}|$ exclusive physical links and $|\mathcal{L}_{st}|$ non-exclusive physical links. Note that the logical link (s, t) uses the non-exclusive link (p_i, q_i) if and only if the edge (s, t) is associated with label L_i in the Minimum Label Spanning Tree problem. Therefore, the set of non-exclusive links used by (s, t) corresponds to the set of labels associated with the edge (s, t) in the Minimum Label Spanning Tree instance. This implies that the set of non-exclusive links used by all logical links in T is exactly the set of labels associated with T in the Minimum Label Spanning Tree problem. Therefore, the logical links in T use a total of $2(n-1)|\mathcal{L}|$ exclusive links and α non-exclusive links. Since T is a logical spanning tree, this set of physical links contains a cross-layer spanning tree. As a result, we have $\beta \leq 2(n-1)|\mathcal{L}| + \alpha$.

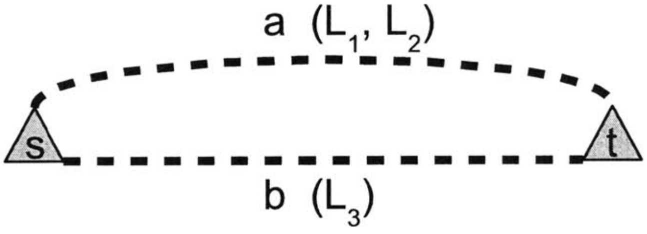


Figure 2-10: Minimum Label Spanning Tree instance.

Now, assume that $\beta < 2(n-1)|\mathcal{L}| + \alpha$. The minimum cross-layer spanning tree S therefore contains less than α non-exclusive links. Let W be the set of logical links that survive if only the phyiscal links in S survive. Since W is a connected subgraph of E_L, it contains a logical spanning tree T that uses less than α non-exclusive links. Since the set of non-exclusive links used by T corresponds to the set of labels associated with the spanning tree T in the Minimum Label Spanning Tree problem, this contradicts the fact that the minimum label spanning tree has α labels. Therefore, we have $\beta \geq 2(n-1)|\mathcal{L}| + \alpha$. □

Because of Lemma 2.14, finding the minimum label spanning tree can be reduced to finding the minimum cross-layer spanning tree under the reduction. □

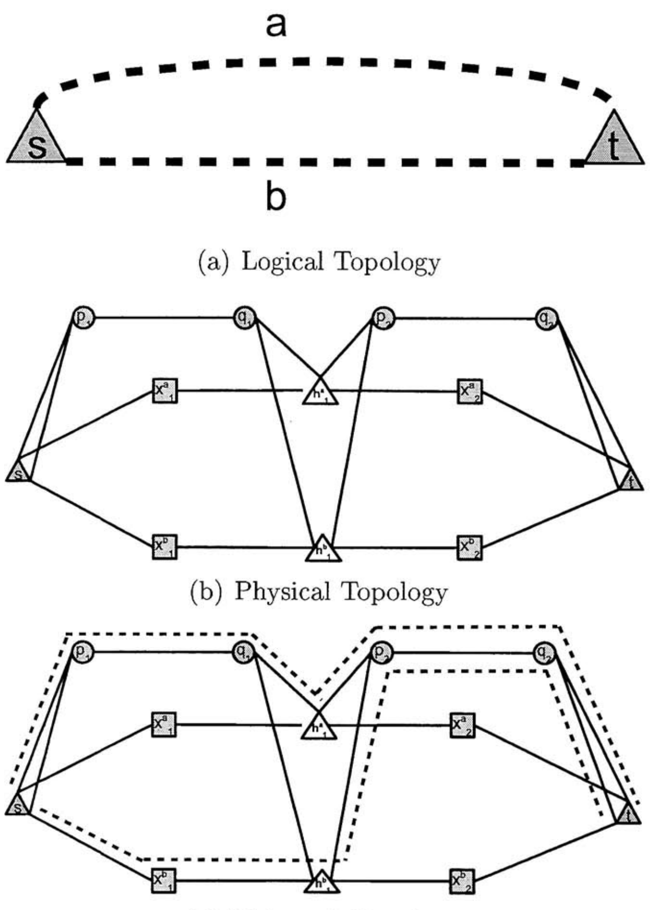
2.7.4 Proof of Theorem 2.9

Theorem 2.9: Computing the Weighted Load Factor for a lightpath routing is NP-Hard even if the weight assignment w_{st} for the logical links is fixed.

Proof. We construct a reduction from the NP-Hard *Uniform Sparsest Cut [7]* problem:

- **Uniform Sparsest Cut:**

  Given an undirected graph $G = (V, E)$, compute the value of $\min_{S \subset V_L} \frac{|\delta(S)|}{|S||V-S|}$.



(a) Logical Topology

(b) Physical Topology

(c) Lightpath Routing

Figure 2-11: Minimum Cross-Layer Spanning Tree instance.

Given the graph $G = (V, E)$ in an instance of Uniform Sparsest Cut problem, we construct an instance of the Weighted Load Factor problem, with the weight assignment w_{st} fixed, such that the optimal values of the two problems are identical. Without loss of generality, we assume G is connected. We will construct a physical topology, logical topology, lightpath routing f_{st}^{ij} and weight assignment w_{st} of the logical links based on the graph $G = (V, E)$ in the Uniform Sparsest Cut instance.

- **Logical Topology:** The logical topology is a complete graph on the vertex set $V_L = V$. Each logical link (s, t) has weight $w_{st} = 1$.

- **Physical Topology:** The physical topology is a complete graph on the vertex set $V_P = V \cup \{u, v\}$, where u and v are two new vertices not in V.

- **Lightpath Routing:** For each logical link (s,t), if (s,t) is an edge of G in the Uniform Sparsest Cut instance, the logical link takes on the physical route $s \to u \to v \to t$. Otherwise, it takes on the physical route $s \to t$.

Let S be an arbitrary subset of V. Let $\delta_{SC}(S)$ be the cut set of S with respect to graph G of the Uniform Sparsest Cut instance, and let $\delta_L(S)$ be the cut set of S with respect to the logical topology G_L, which is a complete graph on $V_L = V$. We claim the following equality:

$$\frac{|\delta_{SC}(S)|}{|S||V-S|} = \max_{(i,j)\in E_P} \frac{\sum\limits_{(s,t)\in\delta_L(S)} w_{st} f_{st}^{ij}}{\sum\limits_{(s,t)\in\delta_L(S)} w_{st}}. \tag{2.12}$$

This is because every physical link not attached to u or v is used by at most one logical link. In addition, any logical link that uses a physical link in the form (x,u) or (v,x), for any x in V_P, also uses (u,v) in the lightpath routing. Since G is connected, for each $S \subset V$, there is at least one logical link in $\delta_{SC}(S)$ that uses the physical link (u,v). Therefore, for any $S \subset V_L$, the physical link (u,v) carries the largest number of logical links in $\delta_L(S)$. Since a logical link uses (u,v) if and only if the corresponding edge exists in G, the number of logical links in $\delta_L(S)$ using (u,v) is $|\delta_{SC}(S)|$. Therefore, the fraction of weight carried by the physical link (u,v) is $\frac{|\delta_{SC}(S)|}{|\delta_L(S)|} = \frac{|\delta_{SC}(S)|}{|S||V-S|}$. This implies the sparsest cut value equals the Weighted Load Factor value. □

2.7.5 Proof of Theorem 2.10

Let $MCLC$ and $MCLC^R$ be the optimal objective values for formulation $\mathsf{M}_{\mathsf{MCLC}}$ and its linear relaxation $\mathsf{M}^{\mathsf{R}}_{\mathsf{MCLC}}$ respectively. And let WLF be the Weighted Load Factor of the lightpath routing. Theorem 2.10 declares the following:

Theorem 2.10: $MCLC^R \leq WLF \leq MCLC$.

Proof. Recall that the ILP formulation for $MCLC$ is:

$$\mathsf{M}_{\mathsf{MCLC}}: \text{Minimize} \sum_{(i,j)\in E_P} y_{ij}, \quad \text{subject to:}$$

$$d_t - d_s \leq \sum_{(i,j)\in E_P} y_{ij} f_{ij}^{st}, \qquad \forall (s,t) \in E_L \tag{2.13}$$

$$\sum_{n\in V_L} d_n \geq 1 \tag{2.14}$$

$$d_0 = 0, \quad d_n, y_{ij} \in \{0,1\}, \qquad \forall n \in V_L, (i,j) \in E_P$$

where f_{ij}^{st} are binary constants such that logical link (s,t) traverses physical fiber (i,j) if and only if $f_{ij}^{st} = 1$.

For the rest of the proof, for any subset S of the logical nodes V_L, we denote $\delta(S)$ to be the cut set of S, i.e., the set of logical links with exactly one end point in S.

We first prove that $MCLC^R \leq WLF$. To do this, we construct the dual [17] of $\mathsf{M}^{\mathsf{R}}_{\mathsf{MCLC}}$:

$$\mathsf{M}^{\mathsf{Dual,R}}_{\mathsf{MCLC}}: \text{Maximize } q, \quad \text{subject to:}$$

$$\sum_{(s,t)\in E_L} g^{st} f_{ij}^{st} \leq 1, \quad \forall (i,j) \in E_P \tag{2.15}$$

$$q + \sum_{(s,t)\in E_L} g^{st} - \sum_{(t,s)\in E_L} g^{ts} \leq 0, \quad \forall s \neq 0 \tag{2.16}$$

$$q, g^{st} \geq 0, \quad \forall (s,t) \in E_L$$

The variables y_{ij} in the primal $\mathsf{M}^{\mathsf{R}}_{\mathsf{MCLC}}$ correspond to Constraint (2.15) in the dual. Similarly, the variables d_s, where $s \neq 0$, in the primal correspond to Constraint (2.16) in the dual. For Constraints (2.13) and (2.14) in the primal, the corresponding variables in the dual are g^{st} and q respectively. We can interpret the variable g^{st} as the flow value assigned to logical link (s,t). Then Constraint (2.15) requires that the total flow on each physical fiber be at most 1. Constraint (2.16) requires at least q units of incoming flow for all nodes other than node 0. Intuitively, the dual program tries to maximize the value q such that the node 0 sends at least q units of flow to every other node, subject to the capacity constraint for each fiber.

We first prove Lemma 2.15, which will be used to establish the lower bound on WLF.

Lemma 2.15 *Let (q, g) be a feasible solution for $\mathsf{M}_{\mathsf{MCLC}}^{\mathsf{Dual,R}}$, and let*

$$g(S) = \sum_{(s,t)\in E_L : s\notin S, t\in S} g^{st} - \sum_{(s,t)\in E_L : s\in S, t\notin S} g^{st}$$

be the net flow into the cut set S. Then $g(S) \geq kq$, for any $S \subseteq V_L \backslash \{0\}$ with $k = |S|$.

Proof. Consider an arbitrary node set $S \subseteq V_L \backslash \{0\}$, and let $k = |S|$. We prove by induction on k that $g(S) \geq kq$.

- **Base case**: $k = 0$: In this case, S is an empty set and $g(S) \geq kq$ trivially.
- **Inductive case**: Suppose for some $0 \leq k < |V_L| - 1, g(S) \geq kq$ for all S with $|S| = k$ and $0 \notin S$. Now let $S^{'}$ be any subset of $k + 1$ nodes that does not contain node 0, let b be an arbitrary node in $S^{'}$, and let $S_b^{'} = S^{'} \backslash \{b\}$. Since $S_b^{'}$ is a set of k nodes, by induction hypothesis, we have $g(S_b^{'}) \geq kq$. It follows that:

$$\begin{aligned} g(S^{'}) &= g(S_b^{'}) + \sum_{(t,b)\in E_L} g^{tb} - \sum_{(b,t)\in E_L} g^{bt} \\ &\geq g(S_b^{'}) + q, \quad \text{by Constraint (2.16)} \\ &\geq (k+1)q. \end{aligned}$$

By induction, $g(S) \geq kq \; \forall S \subseteq V_L \backslash \{0\}$ and $k = |S|$. □

Now we are ready to prove that $MCLC^R \leq WLF$. Given an optimal solution (q^*, g^*) to the formulation $\mathsf{M}_{\mathsf{MCLC}}^{\mathsf{Dual,R}}$, the value of g^{st^*} is a feasible assignment of the variable w_{st} in the Weighted Load Factor formulation $\mathsf{M}_{\mathsf{WLF}}$. The corresponding objective value for this assignment is:

$$\begin{aligned}
&\min_{S\subset V_L,(i,j)\in E_P} \frac{\sum\limits_{(s,t)\in\delta(S)} g^{st^*}}{\sum\limits_{(s,t)\in\delta(S)} g^{st^*} f_{ij}^{st}} \\
\geq &\frac{q^*}{\sum\limits_{(s,t)\in\delta(S)} g^{st^*} f_{ij}^{st}}, && \text{by Lemma 2.15} \\
\geq &q^*, && \text{by Constraint (2.15)}
\end{aligned}$$

which implies $WLF \geq q^*$. On the other hand, by Duality Theorem [17], the optimal value for $\mathsf{M}^{\mathsf{R}}_{\mathsf{MCLC}}$ is exactly q^*. Therefore we have $MCLC^R \leq WLF$.

Next, we prove that $WLF \leq MCLC$. Let C be the set of physical fibers that constitute a Min Cross Layer Cut, and let a be an arbitrary node in the logical network. Let $S_C \subset V_L$ be the set of nodes reachable from a after C has been removed from the physical network. It follows that all logical links in $\delta(S_C)$ use fibers in C.

Let w be the weight function on E_L that achieves the optimal Weighted Load Factor, and let $w(S_C)$ be the total weight of the logical links in $\delta(S_C)$. Also, let (i^*, j^*) be the physical fiber that carries the most weight for lightpaths in $\delta(S_C)$. The definition of WLF implies that:

$$\begin{aligned}
WLF = &\min_{S\subset V_L,(i,j)\in E_P} \frac{\sum\limits_{(s,t)\in\delta(S)} w_{st}}{\sum\limits_{(s,t)\in\delta(S)} w_{st} f_{ij}^{st}} \\
\leq &\frac{\sum\limits_{(s,t)\in\delta(S_C)} w_{st}}{\sum\limits_{(s,t)\in\delta(S_C)} w_{st} f_{i^*j^*}^{st}}. \qquad (2.17)
\end{aligned}$$

Next, since all logical links in $\delta(S_C)$ use fibers in C, we have:

$$\begin{aligned}
\sum_{(s,t)\in\delta(S_C)} w_{st} \leq &\sum_{(i,j)\in C} \sum_{(s,t)\in\delta(S_C)} w_{st} f_{ij}^{st} \\
\leq &|C| \sum_{(s,t)\in\delta(S_C)} w_{st} f_{i^*j^*}^{st}. \qquad (2.18)
\end{aligned}$$

Finally, combining inequalities (2.17) and (2.18), we have:

$$WLF \leq \frac{\sum\limits_{(s,t)\in\delta(S_C)} w_{st}}{\sum\limits_{(s,t)\in\delta(S_C)} w_{st} f_{i^*j^*}^{st}} \leq |C| = MCLC.$$

□

Chapter 3

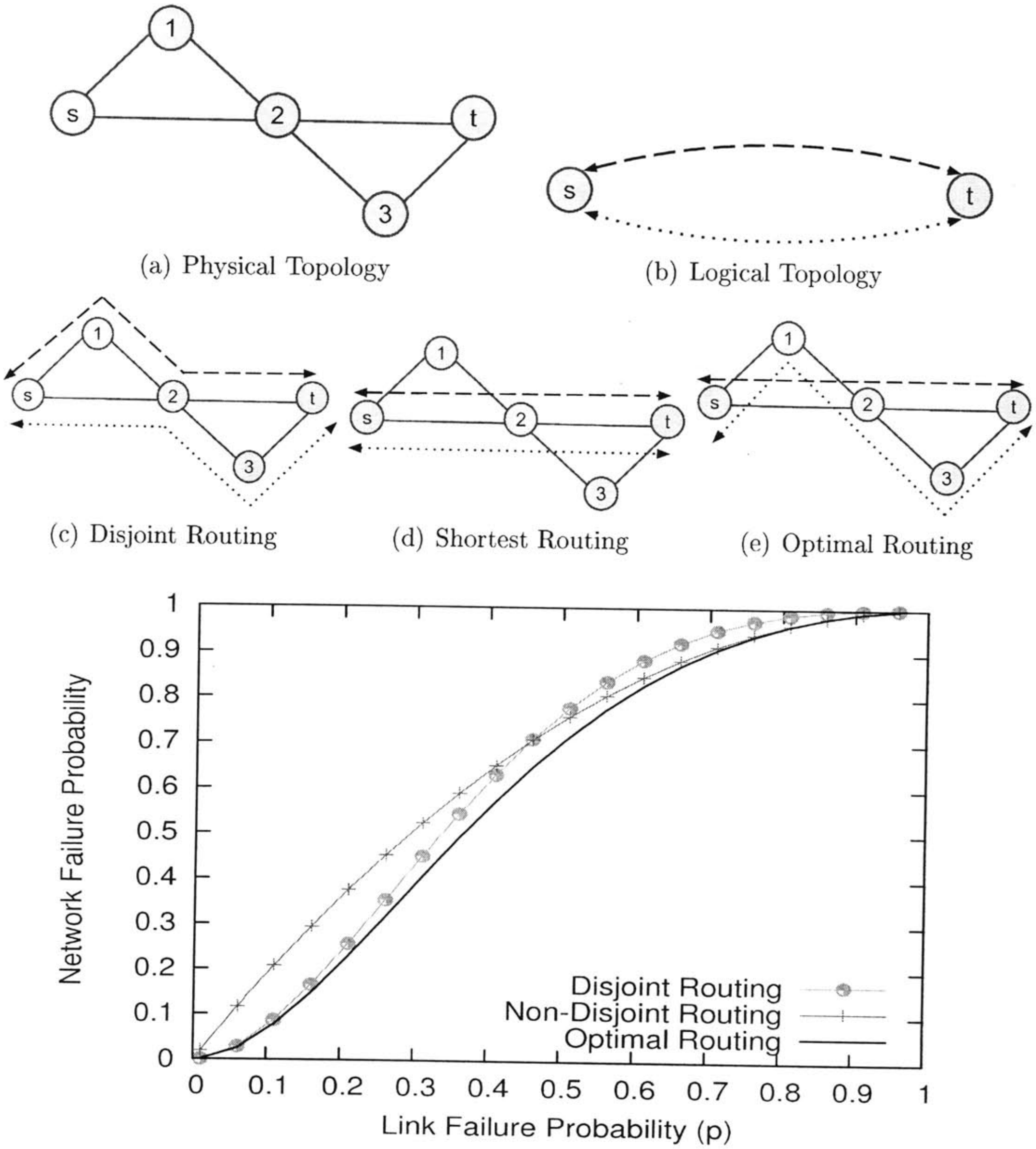
Assessing Reliability for Layered Network under Random Physical Failures

3.1 Introduction

The study of cross-layer survivability in Chapter 2 is based on a deterministic failure model, where survivability is defined by one (smallest) set of physical failures that disconnect the logical topology. In this chapter, we extend our study to the random physical failure model where all physical links fail independently with probability p. This probabilistic failure model represents a snapshot of a network where links fail and are repaired according to some Markovian process. Hence, p represents the steady-state probability that a physical link is in a failed state. The *cross-layer reliability* of the network, defined to be the probability that the logical topology stays connected under the random physical failures, is a natural generalization of the single-layer all-terminal reliability, which has been extensively studied in the literature (see [32] for example). However, as shown in the previous chapter, the structural properties in layered networks are significantly different from single-layer networks. This makes many of the existing approaches either inapplicable or inefficient in the multi-layer setting. In particular, in additional to the physical and logical topologies, the underlying lightpath routing of a layered network determines the way the logical network is affected by the physical failures, and therefore plays an important role in the overall reliability of the network.

For example, in Figure 3-1, the logical topology consists of two parallel links between nodes s and t. Suppose every physical link fails independently with probability p. The first lightpath routing in Figure 3-1(c) routes the two logical links using link-disjoint physical paths $(s, 1, 2, t)$ and $(s, 2, 3, t)$. Under this routing, the logical network will be disconnected with probability $(1-(1-p)^3)^2$. On the other hand, the second lightpath routing in Figure 3-1(d), which routes the two logical links over the same shortest physical route $(s, 2, t)$, has failure probability $2p - p^2$. While disjoint path routing is generally considered more reliable, it is only true in this example for small values of p. For large p (e.g. $p > 0.5$), the second lightpath routing is actually more reliable. Therefore, whether one lightpath routing is better than another may depend on the value of p. In some cases, there may exist a lightpath routing with lower failure probability over all values of p, as shown in Figure 3-1(e).

Therefore, in order to design a reliable layered network, it is important to develop a better understanding of the role of lightpath routings in cross-layer reliability. To achieve this, we will extend the polynomial expression for single-layer network reliability to the layered setting. In Section 3.3 we define the *cross-layer failure polynomial*, which provides a formula for network reliability as a function of the link failure probability. Hence, the cross-layer reliability can be estimated by approximating the coefficients of the polynomial. Exploiting this relationship, in Sections 3.4-3.7 we develop Monte Carlo based estimation methods that approximates cross-layer reliability with provable accuracy. Our method is not tailored to a particular probability of link failure, and consequently, it does not require resampling in order to estimate reliability under different values of link failure probability. That is, once the polynomial is estimated, it can be used for any value of link failure probability without resampling. Our approach is immediately applicable to single-layer networks as well.



(a) Physical Topology

(b) Logical Topology

(c) Disjoint Routing

(d) Shortest Routing

(e) Optimal Routing

Figure 3-1: Example of disjoint, shortest and optimal routings: Non-disjoint routings can sometimes be more reliable than disjoint routings. Optimally reliable routings over all values of p sometimes exist.

Another interesting property of the polynomial expression for reliability is that its coefficients contain the structural information of the cross-layer topology, especially lightpath routing. Consequently, it gives clear insights on how lightpath routing should be designed for better reliability. This, together with our estimation algorithm, enables us to revisit the network design problem from the viewpoint of network reliability. In Section 3.8 we will investigate the connection between cross-layer reliability and Min Cross Layer Cut, the survivability metric used in Chapter 2, and study the performance of the lightpath routing algorithms presented in Chapter 2 under this random failure model. We will briefly discuss several extensions to our failure model in Section 3.9, and how the reliability estimation algorithms can be applied to these new settings. The insights developed in this chapter, in particular, the study of the failure polynomial, lays the groundwork for our studies in the next two chapters, which focus on designing networks to maximize reliability.

In Appendix 3.11.3, we briefly discuss an alterative approach based on importance sampling [97] to assess reliability of layered networks, and constrast it with our failure polynomial approach.

3.2 Previous Work

The network reliability estimation problem has been extensively studied in the single-layer setting. Valiant [114] first showed that computing reliability in the single-layer setting is $\#P$-complete[1]. Provan and Ball [87] later showed that it is $\#P$-complete even to approximate the reliability up to ϵ relative accuracy. Due to the inherent complexity, most of the previous works in this context focused on approximating the actual reliability. Although there are some works aimed at exact computation of reliability through graph transformation and reduction [27, 73, 83, 86, 98, 106, 107, 111], the applications of such methods are highly limited since they are targeted to particular topologies. Furthermore, those methods cannot be used for estimating cross-layer reliability because they assume independence between link failures, while failures are often correlated in multi-layer networks.

Monte Carlo simulation was also used for estimating the single-layer reliability for some fixed link failure probability. Using simulation, the reliability can be ap-

[1] The complexity class #P is the counting equivalent of NP. While a decision problem in NP asks about whether a feasible solution exists subject to certain constraints, its corresponding problem in #P asks about *how many* of such feasible solutions exist.

A problem is #P-complete if and only if it is in #P, and every problem in #P can be reduced to it in polynomial-time. An algorithm that solves a #P-complete problem in polynomial time will imply P=NP, and is therefore unlikely to exist.

proximated to an arbitrary accuracy, but the number of iterations required by direct simulation tends to be very large when the failure probability is small. There are various algorithms designed specifically to optimize for this case [41,42,62,63]. However, each run of these algorithms only estimates the reliability for a given link failure probability; and the algorithm must be repeated for a different failure probability.

Another approach is to use a polynomial expression for reliability [12] and estimate every coefficient appearing in the polynomial; where the reliability can be approximated using the estimated coefficients. The advantage of this approach over simulation is that once every coefficient is estimated, they can be used for any value of failure probability. Most of the works in this context have focused on bounding the coefficients by applying subgraph counting techniques and results from combinatorics [24,31,53,89,94]. This approach is computationally attractive, but its estimation accuracy is not guaranteed. Some previous works studied the regime of low failure probability by focusing on small cut sets [2,16]. In [82], a random sampling technique is used to enhance those bounding results. In particular, [82] considers another form of the polynomial used in [13], and estimates some of the coefficients by enumerating spanning trees in the graph. These estimates are used to improve the algebraic bound in [13]. This approach is relevant to our work in that it tries to approximate the coefficients in the polynomial through random sampling. However, the algorithm proposed in [82] is based on sampling spanning trees in the network, which is not immediately applicable to our multi-layer setting because the properties of cross-layer spanning trees is vastly different from their single-layer counterparts; and sampling minimum spanning trees in layered networks becomes a much more difficult problem, as discussed in Section 2.2.3.

In this chapter, we take a different approach from [82] by sampling cross-layer cuts. Even though finding minimum cross-layer cuts is an NP-Hard problem, our cut-based approach is feasible in the cross-layer setting due to the following reasons:

1. The size of minimum cross-layer cut is bounded above by the minimum logical node degree, which is usually a constant. In practice, it is often easier to find or enumerate minimum cross-layer cuts than spanning-trees, which is lower

bounded by the number of logical nodes, as shown in Theorem 2.4.

2. Except for cross-layer cuts of small size, it can be shown that cross-layer cuts are abundant in a layered network in general. This makes cut sampling a promising approach.

In Section 3.4, we will develop a reliability estimation algorithm based on the above insight. Before that, we first formally describe our model and provide some mathematical background.

3.3 Model and Background

A multi-layer network is modelled by a logical topology $G_L = (V_L, E_L)$ built on top of the physical topology $G_P = (V_P, E_P)$ through a lightpath routing, where V and E are the set of nodes and links respectively. The lightpath routing is denoted by $f = [f_{ij}^{st}, (i,j) \in E_P, (s,t) \in E_L]$, where f_{ij}^{st} takes the value 1 if logical link (s,t) is routed over physical link (i,j), and 0 otherwise.

We consider a random failure model where the state of each physical link $(i,j) \in E_P$ is represented by the 0-1 random variable x_{ij}, which equals 0 if and only if the physical link (i,j) fails. Let $\mathcal{H} = 2^{E_P}$ be the family of all subsets of the physical links E_P. We define a *network state* $S \in \mathcal{H}$ as the set of physical links that fail, that is, $S = \{(i,j) : x_{ij} = 0\}$.

Each physical link fails independently with probability p. If a physical link (i,j) fails, all the logical links (s,t) carried over (i,j) (i.e., (s,t) such that $f_{ij}^{st} = 1$) also fail. A network state S is called a *cross-layer cut* if and only if the failure of the physical links in S causes the logical network to be disconnected. Let R be a 0-1 random variable on $\mathcal{H}$ such that $R(S) = 1$ if and only if S is not a cross-layer cut. Then, the *reliability* of the layered network is defined to be $Pr(R = 1)$. Similarly, the *unreliability* is defined to be $Pr(R = 0)$.

3.3.1 Cross-Layer Failure Polynomial

Since cross-layer reliability generalizes all-terminal reliability in single-layer networks, the results by Valiant [114] and Provan et. al. [87] immediately imply that approximating cross-layer reliability within a constant factor is $\#P$-Complete. Hence, our goal in this chapter is to develop a probabilistic algorithm that can accurately estimate the reliability with high probability. As we discussed in Section 3.1, the relative reliability performance among lightpath routings depend heavily on the value of p. Therefore, when comparing lightpath routings, it is often necessary to assess the reliability at different link failure probabilities in order to obtain better insight from the comparison. For this purpose, it is useful to develop an estimation method such that once an estimation is made, the result can be used for every value of p. Therefore, we will develop an algorithm that outputs the reliability approximation as a polynomial in p, so that comparing different lightpath routings at different link failure probabilities is trivial. As we will see in Section 3.8, the failure polynomial also provides important insights to the design of lightpath routings for better reliability.

The polynomial expression for reliability presented here is a natural extension of the single-layer polynomial [12] to the cross-layer setting. Assume that there are m physical links, i.e., $|E_P| = m$. The probability associated with a network state S with exactly i physical link failures (i.e., $|S| = i$) is $p^i(1-p)^{m-i}$. Let N_i be the number of cross-layer cuts S with $|S| = i$, then the probability that the network gets disconnected is simply the sum of the probabilities over all cross-layer cuts, i.e.,

$$F(p) = \sum_{i=0}^{m} N_i p^i (1-p)^{m-i}. \tag{3.1}$$

Therefore, the failure probability of a multi-layer network can be expressed as a polynomial in p. The function $F(p)$ will be called *cross-layer failure polynomial* or simply the *failure polynomial.* The vector $[N_0, \ldots, N_m]$ plays an important role in assessing the reliability of a network. In particular, one can simply plug the value of p in the above failure polynomial to compute the reliability if the values of N_i are known.

Intuitively, each N_i represents the number of cross-layer cuts of size i in the network. Clearly, if $N_i > 0$, then $N_j > 0, \forall j > i$ (because any cut of size i will still be a cut with the addition of more failed links). The smallest i such that $N_i > 0$ is of special importance because it represents the Min Cross Layer Cut (MCLC) of the network, i.e., it is the minimum number of physical link failures needed to disconnect the logical network. Although computing the MCLC is NP-Hard [70], for practical purposes, the MCLC of a network is typically upper bounded by some constant, such as the minimum node degree of the logical network. Therefore, for the rest of the chapter, we denote the MCLC value of the network by d, and assume that it is a constant independent of the physical network size. It is important to note that $N_i = 0, \forall i < d$, and the term $N_d p^d (1-p)^{m-d}$ in the failure polynomial dominates for small values of p. Consequently, if a lightpath routing tries to maximize MCLC, i.e., make d as large as possible, it will achieve good reliability in the low failure probability regime. On the other hand, its reliability performance is not guaranteed in other regimes. This will be further discussed in Section 3.8, where we study the reliability performance of the lightpath routing algorithms presented in Chapter 2. A similar observation was made for single-layer networks in [20].

In this chapter, we focus on approximating the failure polynomial. We will use the following notions of approximation.

Definition 3.1 (Relative Approximation) *A function $\hat{F}(p)$ is an ϵ-approximation for the failure polynomial $F(p)$ if*

$$|F(p) - \hat{F}(p)| \leq \epsilon F(p), \textit{ for all } p \in [0,1].$$

This relative error is typically the measure of interest in the literature of reliability estimation. However, as mentioned above, it is also $\#P$-complete to approximate the reliability to ϵ accuracy [87]. Hence, it is not likely that there exists a *deterministic* ϵ-approximation algorithm requiring reasonably low computation. For this reason, our estimation focuses on the following probabilistic approximation.

Definition 3.2 ((ϵ, δ)-approximation) *A function $\hat{F}(p)$ is an (ϵ, δ)-approximation* *for the failure polynomial $F(p)$ if*

$$\Pr\left[|F(p) - \hat{F}(p)| \leq \epsilon F(p)\right] \geq (1-\delta), \text{ for all } p \in [0,1].$$

In other words, an (ϵ, δ)-approximation algorithm approximates the polynomial to ϵ relative accuracy with high probability. In Sections 3.4 and 3.5, we will present randomized (ϵ, δ)-approximation algorithms for the failure polynomial.

3.3.2 Monte Carlo Simulation

Our estimation algorithm is based on Monte Carlo simulation techniques. The central theme of such Monte Carlo techniques is based on the *Estimator Theorem*, presented below. Let U be a ground set defined as the set of all possible events (e.g., all network states), and G be a subset of U (e.g., cross-layer cuts). Suppose that we want to estimate $|G|$. To do this, the Monte Carlo method samples an element e from U uniformly at random for T times. For each iteration i, let X_i be the 0-1 random variable that equals 1 if and only if the sampled element $e \in G$. Then the random variable $Y = \frac{|U|\sum_{i=1}^{T} X_i}{T}$ is an unbiased estimator of $|G|$. The Estimator Theorem states that:

Theorem 3.1 (Estimator Theorem [77]) *Let $\rho = \frac{|G|}{|U|}$. Then $Y = \frac{|U|\sum_{i=1}^{T} X_i}{T}$ is an (ϵ, δ)-approximation to $|G|$, provided that*

$$T \geq \frac{4}{\epsilon^2 \rho} \ln \frac{2}{\delta}.$$

In other words, if we sample from the ground set U frequently enough, we can estimate $|G|$ accurately with high probability. According to Theorem 3.1, the ratio ρ, called the *density* of the set G, is inversely proportional to the required sample size T. This is because the squared coefficient of variation of Y, defined as $\frac{Var(Y)}{E(Y)^2}$, equals $\frac{(1-\rho)}{T\rho}$. Therefore, a sample size T in the order of $\frac{1}{\rho}$ is needed, so that the squared coefficient of variation will not grow with $\frac{1}{\rho}$, which is necessary to keep the relative error small [97].

In the following sections, we will define the sets G and U in various ways to ensure high ρ value, and propose polynomial-time Monte Carlo methods to compute approximations of the failure polynomial.

3.4 Estimating Cross-Layer Reliability

The most straightforward Monte-Carlo method to estimate network reliability is via direct simulation, that is, collect T samples from the universe of network states $\mathcal{H}$, where each sample is obtained by simulating each physical link failure with probability p. For each sample S_i, compute the value R_i of the random variable $R(S_i)$. An unbiased estimator for the reliability is then given by $\frac{\sum_{i=1}^{T} R_i}{T}$. However, such an approach has the following drawbacks:

1. The output of the algorithm is the reliability value for a particular link failure probability p. To assess reliability at a different link failure probability p, a new round of sampling is required.

2. The unreliability of the network R can be arbitrarily small if the link failure probability p is sufficiently small. Therefore, the number of samples required to keep relative error small, which is in the order of $\frac{1}{R}$, can be arbitrarily large.

Our approach to approximating the cross-layer failure polynomial is to estimate the values of N_i in Equation (3.1) separately. If we can estimate each N_i with sufficient accuracy, we will obtain an approximate failure polynomial for the multi-layer network. The idea is formalized in the following theorem.

Theorem 3.2 *Let $\hat{N}_i$ be an ϵ-approximation of N_i for all $i \in \{1, \ldots, m\}$, then the function $\hat{F}(p) = \sum_{i=0}^{m} \hat{N}_i p^i (1-p)^{m-i}$ is an ϵ-approximation for the failure polynomial.*

Proof. For all $0 \leq p \leq 1$,

$$\begin{aligned}|\hat{F}(p) - F(p)| &\leq \sum_{i=0}^{m} |(\hat{N}_i - N_i)| p^i (1-p)^{m-i} \\ &\leq \sum_{i=0}^{m} \epsilon N_i p^i (1-p)^{m-i} \\ &= \epsilon F(p).\end{aligned}$$

□

Corollary 3.3 *Let* A *be an algorithm that computes an* $(\epsilon, \frac{\delta}{m+1})$*-approximation for each* N_i*. Then* A *gives an* (ϵ, δ)*-approximation algorithm for the failure polynomial.*

Proof. By the union bound, the probability that all the N_i estimates are ϵ-approximate is at least $1 - \sum_{i=0}^{m} \frac{\delta}{m+1} = 1 - \delta$. By Theorem 3.2, A gives an (ϵ, δ)-approximation algorithm for the failure polynomial. □

Note that this approach can be considered as a form of *stratified sampling* [97], where the sample space $\mathcal{H}$ is partitioned into multiple subgroups $\mathcal{H}_i$ and the conditional expectations $E[R|\mathcal{H}_i]$ are estimated independently. The expectation of the random variable R is thus given by:

$$E[R] = \sum_{\mathcal{H}_i} E[R|\mathcal{H}_i] Pr(\mathcal{H}_i).$$

For the cross-layer reliability estimation problem, we define each subgroup $\mathcal{H}_i$ to be all possible subsets of E_P with size i, that is, $\mathcal{H}_i = \{S \in E_P : |S| = i\}$. It follows that $Pr(\mathcal{H}_i) = \binom{m}{i} p^i (1-p)^{m-i}$, and the conditional expectation, $E[R(S) = 0 | S \in \mathcal{H}_i]$, is simply $\frac{N_i}{\binom{m}{i}}$. The key observation is that the conditional probability and variance is independent of the link failure probability p. As a result, the confidence interval obtained by simulating the conditional events within a subgroup is independent of p. This ensures the effectiveness of the algorithm even if p is small.

As a result of Corollary 3.3, it suffices to obtain a $(\epsilon, \frac{\delta}{m+1})$-approximation for each N_i. In the remainder of this section, we will discuss how this can be achieved.

3.4.1 Estimating N_i

Let $\mathcal{H}_i$ be the family of all subsets of E_P with exactly i physical links. Clearly, N_i is the number of subsets in $\mathcal{H}_i$ that are cross-layer cuts. Hence, one can compute the exact value of N_i by enumerating all subsets in $\mathcal{H}_i$ and counting the number of cross-layer cuts. However, the number of subsets to enumerate is $\binom{m}{i}$, which can be prohibitively large.

An alternative approach to estimating N_i is to carry out Monte Carlo simulation on $\mathcal{H}_i$. Suppose we sample uniformly at random from $\mathcal{H}_i$ for T times, and count the number of cross-layer cuts W in the sample. The Estimator Theorem guarantees that $\binom{m}{i}\frac{W}{T}$ is an $(\epsilon, \frac{\delta}{m+1})$-approximation, provided that:

$$T \geq \frac{4}{\epsilon^2 \rho_i} \ln \frac{2(m+1)}{\delta}, \tag{3.2}$$

where $\rho_i = \frac{N_i}{\binom{m}{i}}$ is the density of cross-layer cuts in $\mathcal{H}_i$. The main issue here is that the exact value for ρ_i, which depends on N_i, is unknown to us. However, if we substitute ρ_i in Equation (3.2) with a lower bound of ρ_i, the number of iterations will be guaranteed to be no less than the required value. Therefore, it is important to establish a good lower bound for ρ_i in order to keep the number of iterations small while achieving the desired accuracy.

3.4.2 Lower Bounding ρ_i

Given a layered network, suppose its Min Cross Layer Cut value d is known, Theorem 3.4 gives a lower bound on ρ_i:

Theorem 3.4 *For* $i \geq d, \rho_i \geq \frac{\binom{m-d}{i-d}}{\binom{m}{i}}$.

Proof. Since d is the Min Cross Layer Cut value, there exists a cross-layer cut S with size d. Any superset of S with i physical links is therefore also a cross-layer cut. Since there are a total of $\binom{m-d}{i-d}$ such supersets, we have $N_i \geq \binom{m-d}{i-d}$, and the theorem follows immediately. □

Therefore, we can use $\tilde{\rho}_i = \frac{\binom{m-d}{i-d}}{\binom{m}{i}}$ as the lower bound for ρ_i in (3.2) to estimate N_i, with the following observations:

1. The MCLC value d needs to be known in advance.

2. The number of iterations can be very large for small values of i. For example, when $i = d$, the number of iterations T required is $\frac{4\binom{m}{d}}{\epsilon^2} \ln \frac{2(m+1)}{\delta}$, which is no better than enumerating all sets in $\mathcal{H}_d$ by brute force.

3. The lower bound $\tilde{\rho}_i$ increases with i. In particular, $\frac{\tilde{\rho}_{i+1}}{\tilde{\rho}_i} = 1 + \frac{d}{i+1-d}$. Therefore, the number of iterations required to estimate N_i decreases with i.

In the next subsection, we will present an algorithm that combines the enumeration and Monte Carlo methods to take advantage of their different strengths. In Section 3.5, we will present enhanced versions of the algorithm which significantly reduces the number of iterations by establishing a much tighter lower bound on ρ_i. The final outcome is an (ϵ, δ)-approximation algorithm for the failure polynomial $F(p)$ that requires only a polynomial number of iterations.

3.4.3 A Combined Enumeration and Monte Carlo Approach

Recall that N_i can be estimated with two different approaches, brute-force enumeration and Monte Carlo. The two approaches can be combined to design an efficient (ϵ, δ)-approximation algorithm for the failure polynomial.

The key observation for the combined approach is that brute-force enumeration works well when i is small, and the Monte Carlo method works well when i is large. Therefore, it makes sense to use the enumeration method to find the Min Cross Layer Cut value d, as well as the associated value N_d. Once we obtain the value of d, we can decide on the fly whether to use the enumeration method or the Monte Carlo method to estimate each N_i, by comparing the number of iterations required by each method.

3.4.4 Time Complexity Analysis

The total number of iterations of this combined approach will be:

$$\sum_{i=0}^{d}\binom{m}{i}+\sum_{i=d+1}^{m}\min\left\{\binom{m}{i},\frac{4\binom{m}{i}}{\epsilon^2\binom{m-d}{i-d}}\ln\frac{2(m+1)}{\delta}\right\},$$

where the terms inside the min operator are the number of iterations required by enumeration and Monte Carlo methods respectively. The total number of iterations can be upper bounded as follows:

$$\begin{aligned}
&\sum_{i=0}^{d}\binom{m}{i}+\sum_{i=d+1}^{m}\min\left\{\binom{m}{i},\frac{4\binom{m}{i}}{\epsilon^2\binom{m-d}{i-d}}\ln\frac{2(m+1)}{\delta}\right\}\\
\leq&(m+1)^d+\frac{4}{\epsilon^2}\ln\frac{2(m+1)}{\delta}\sum_{i=d+1}^{m}\frac{\binom{m}{i}}{\binom{m-d}{i-d}}\\
=&(m+1)^d+\frac{4}{\epsilon^2}\ln\frac{2(m+1)}{\delta}\sum_{i=d+1}^{m}\frac{\binom{m}{d}}{\binom{i}{d}}\\
\leq&(m+1)^d+\frac{4\binom{m}{d}}{\epsilon^2}\ln\frac{2(m+1)}{\delta}\sum_{i=1}^{m-d}\frac{d!}{i^d}\\
\leq&(m+1)^d+\frac{4\binom{m}{d}d!}{\epsilon^2}\ln\frac{2(m+1)}{\delta}\sum_{i=1}^{m-d}\frac{1}{i^d}\\
=&\begin{cases} O(m\log^2 m), & \text{if } d=1\\ O(m^d\log m), & \text{if } d\geq 2,\end{cases}
\end{aligned}$$

where the first inequality is implied by the following lemma:

Lemma 3.5 $\sum_{i=0}^{d}\binom{m}{i}\leq(m+1)^d$.

Proof.

$$\begin{aligned}(m+1)^d - \sum_{i=0}^{d} \binom{m}{i} &= \sum_{i=0}^{d} \left[\binom{d}{i} m^i - \binom{m}{i} \right] \\ &\geq \sum_{i=0}^{d} \left[m^i - \binom{m}{i} \right] \\ &\geq 0\end{aligned}$$

□

Therefore, the algorithm only needs a polynomial number of iterations overall. The improvement in running time of this combined approach is illustrated by Figure 3-2.

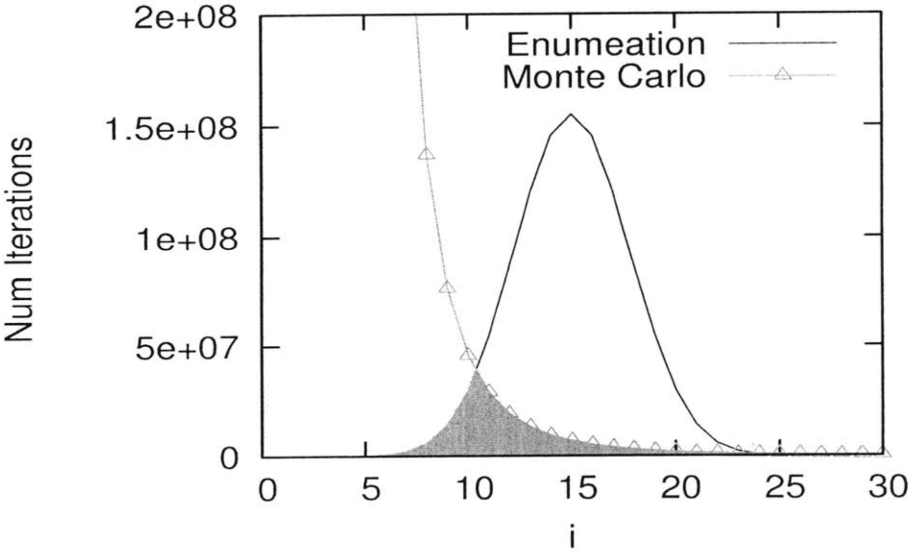


Figure 3-2: Monte-Carlo vs Enumeration: Number of iterations for estimating N_i, for a network with 30 physical links, $\epsilon = 0.01, \delta = \frac{0.01}{31}, d = 4$. The shaded region represents the required iterations for the combined approach.

3.5 Improved ρ_i Lower Bounds for Reliability Estimation

The running time performance of the algorithm introduced in the previous section hinges on the tightness of the lower bounds ρ_i used for the algorithm. In this section, we discuss ways to tighten the lower bounds.

The idea behind these improved bounds is based on the observation that any superset of a cross-layer cut is also a cross-layer cut. Let $\mathcal{F} = \{C_1, \ldots, C_n\}$ be a collection of cross-layer cuts. For each $C_j \in \mathcal{F}$, let $\partial_i(C_j) \subseteq \mathcal{H}_i$ be the family of supersets of C_j with i physical links. Similarly, let $\partial_i(\mathcal{F}) = \cup_{C_j \in \mathcal{F}} \partial_i(C_j)$ be the union over all $\partial_i(C_j)$. Using the terminology in [22], the family of subsets $\partial_i(\mathcal{F})$ is called the i^{th} *upper shadow* for $\mathcal{F}$. The following theorem provides a lower bound on ρ_i in terms of ∂_i:

Theorem 3.6 *Let $\mathcal{F}$ be a collection of cross-layer cuts with size less than i, then* $\rho_i \geq \frac{|\partial_i(\mathcal{F})|}{\binom{m}{i}}$.

Proof. Every set $S \in \partial_i(\mathcal{F})$ is a superset of the some cross-layer cut in $\mathcal{F}$, and is therefore a cross-layer cut with size i. Therefore, $\partial_i(\mathcal{F})$ is a collection of cross-layer cuts with size i, which implies $|\partial_i(\mathcal{F})| \leq N_i$. It follows that $\frac{|\partial_i(\mathcal{F})|}{\binom{m}{i}} \leq \frac{N_i}{\binom{m}{i}} = \rho_i$. □

Therefore, if we know the value of $|\partial_i(\mathcal{F})|$, we can use $\frac{|\partial_i(\mathcal{F})|}{\binom{m}{i}}$ as the lower bound for ρ_i in the Monte Carlo method to estimate N_i. Note that if $\mathcal{F}$ contains only a Min Cross Layer Cut of the network, the value of $\frac{|\partial_i(\mathcal{F}|}{\binom{m}{i}}$ is equal to the bound given by Theorem 3.4. Therefore, Theorem 3.6 generalizes the lower bound result in Section 3.4.2.

Although the value of each $|\partial_i(C_j)| = \binom{m-|C_j|}{i-|C_j|}$ can be computed easily, finding the size of the union $\partial_i(\mathcal{F}) = \cup_{C_j \in \mathcal{F}} \partial_i(C_j)$ can be difficult because the sets $\partial_i(C_j)$ are not disjoint. Instead of computing $|\partial_i(\mathcal{F})|$ precisely, we introduce techniques for lower-bounding $|\partial_i(\mathcal{F})|$. The first technique, introduced in Section 3.5.1, is based on importance sampling for the Union of Sets problem [64]. The second technique, introduced in Section 3.5.2, is to bound the size of $\partial_i(\mathcal{F})$ with a recursive formula, based on the Kruskal-Katona Theorem [22].

3.5.1 Lower Bound by Approximating Union of Sets

Given a set of cross-layer cuts $\mathcal{F}$, the problem of estimating the size of its upper shadow $\partial_i(\mathcal{F})$, can be formulated as the Union of Sets Problem [64], for which a Monte-Carlo based approach exists using the technique of importance sampling. We summarize the result in this section and leave the detailed proofs in Appendix 3.11.1.

Theorem 3.7 *Let $\mathcal{F} = \{C_1, \ldots, C_n\}$ be a collection of cross-layer cuts of the layered network. For each $C_j \in \mathcal{F}$, let $\partial_i(C_j)$ be the i^{th} upper shadow of C_j. There exists a Monte Carlo method that produces an $(\epsilon_{lb}, \delta_{lb})$-approximation, $\hat{L}_i$, for $L_i = |\partial_i(\mathcal{F})|$, provided that the number of samples is at least:*

$$T_{lb} = \frac{4|\mathcal{F}|}{\epsilon_{lb}^2} \ln \frac{2}{\delta_{lb}}. \tag{3.3}$$

Proof. Let $U = \{(S, j) : j \in \{1, \ldots, |\mathcal{F}|\}, S \in \partial_i(C_j)\}$ be the ground set for the Monte Carlo algorithm, and let $G = \{(S, j) : S \in \partial_i(\mathcal{F}), j = \min\{k : S \in \partial_i(C_k)\}\}$ be the events of interest. We show in Appendix 3.11.1 that the ground set U can be sampled uniformly at random. Since $|G| = |\partial_i(\mathcal{F})|$ and $\frac{|G|}{|U|} \geq \frac{1}{|\mathcal{F}|}$, Theorem 3.7 follows immediately from the Theorem 3.1. □

Theorem 3.7 implies $\hat{\rho}_i = \frac{\hat{L}^i}{(1+\epsilon_{lb})\binom{m}{i}}$ is a lower bound on ρ_i with probability at least $1 - \delta_{lb}$. The following theorem describes how such a probabilistic lower bound can be used to estimate N_i.

Theorem 3.8 *Let $\hat{L}_i$ be an $(\epsilon_{lb}, \delta_{lb})$-approximation for $|\partial_i(\mathcal{F})|$. Then, the Monte Carlo method described in Section 3.4.1 yields an $(\epsilon_{mc}, \delta_{lb} + \delta_{mc})$-approximation for*

N_i, provided that the number of samples is at least:

$$T_{mc} = \frac{4(1+\epsilon_{lb})\binom{m}{i}}{\epsilon_{mc}^2 \hat{L}_i} \ln \frac{2}{\delta_{mc}}. \tag{3.4}$$

Proof. By definition of $\hat{L}_i$, the probability that $\hat{\rho}_i = \frac{\hat{L}_i}{(1+\epsilon_{lb})\binom{m}{i}}$ is *not* a lower bound on ρ_i is at most δ_{lb}. Given that $\hat{\rho}_i$ is a lower bound for ρ_i, by the Estimator Theorem, the probability that $\hat{N}_i$ is *not* an ϵ_{mc}-approximation for N_i is at most δ_{mc}. Hence, by the union bound, the probability that none of these "bad" events happen is at least $1-(\delta_{lb}+\delta_{mc})$, and the theorem follows. □

To apply this result to reliability estimation, we can modify our algorithm presented in Section 3.4.3 to also maintain the collection $\mathcal{F}$ of cross-layer cuts as we carry out the enumeration or Monte Carlo methods. Specifically, as we discover a cross-layer cut C_j with size i when estimating N_i, we will add the cut C_j to our collection $\mathcal{F}$. When we move on to estimate N_{i+1}, we will have a collection $\mathcal{F}$ of cross-layer cuts with size i or smaller. We can therefore apply Theorem 3.6 to obtain a lower bound for N_{i+1}. Note that the size of $\partial_i(\mathcal{F})$ is monotonic in $\mathcal{F}$. Therefore, the more cross-layer cuts that are included in $\mathcal{F}$, the better the lower bound is.

3.5.2 Lower Bound based on Kruskal-Katona Theorem

We can also derive a lower bound on ρ_i based on the values of N_j for $j < i$, using the Kruskal-Katona theorem. Let $[m] = \{1, \dots, m\}$, i.e., $[m]$ is the enumeration of physical links. Let $\mathcal{H}_i^m = \{S \subseteq [m] : |S| = i\}$ be a family of subsets of $[m]$ with size i. For any $\mathcal{F} \subseteq \mathcal{H}_j^m$ with $j < i$, we denote $\partial_i^m(\mathcal{F})$ to be the i^{th} upper shadow over $[m]$ for $\mathcal{F}$.

We define the *lexicographic ordering* on $\mathcal{H}_i^m$ as follows: Given any two subsets S_1 and S_2 in $\mathcal{H}_i^m$, S_1 is *lexicographically smaller than* S_2 if and only if $\min\{i : i \in S_1 \Delta S_2\} \in S_1$, where Δ denotes the symmetric difference between the two sets, i.e., $S_1 \Delta S_2 =$

$S_1 \cup S_2 - S_1 \cap S_2$. For example, the set $\{1, 2, 4\}$ is lexicographically smaller than $\{1, 3, 4\}$ because the smallest element where the two sets differ, 2, is in the first set.

Given $\mathcal{H}_i^m$, the family of all subsets with size i, let $\mathcal{H}_i^m(k) \subseteq \mathcal{H}_i^m$ be the first k elements of $\mathcal{H}_i^m$ under the lexicographical ordering. The Kruskal-Katona theorem states that $\mathcal{H}_i^m(k)$ yields the smallest upper shadow among all k-subset of $\mathcal{H}_i^m$:

Theorem 3.9 ([22]) *For any $i < j$ and $\mathcal{F} \subseteq \mathcal{H}_i^m$,*

$$|\partial_j^m(\mathcal{H}_i^m(|\mathcal{F}|))| \leq |\partial_j^m(\mathcal{F})|. \tag{3.5}$$

In other words, for a fixed value of k, the upper shadow for $\mathcal{F}$ with $|\mathcal{F}| = k$ is minimized if $\mathcal{F}$ consists of the first k subsets of $\mathcal{H}_i^m$ in lexicographical order. Therefore, suppose a multi-layer network has a N_i cross-layer cuts with size i, Theorem 3.9 implies that $N_j \geq |\partial_j^m(\mathcal{H}_i^m(N_i))|$ for all $j > i$. We prove the following recursive formula for $|\partial_j^m(\mathcal{H}_i^m(N_i))|$:

Theorem 3.10 *For $i < j \leq m$ and $1 \leq k \leq \binom{m}{i}$, let $w = \max\left\{0 \leq r < i : \binom{m-r}{i-r} \geq k\right\}$. Also, let $t = m - (w+1), u = j - (w+1)$ and $v = i - (w+1)$. Then:*

$$|\partial_j^m(\mathcal{H}_i^m(k))| = \begin{cases} \binom{m-i}{j-i}, & \textit{if } k = 1 \\ \binom{t}{u} + |\partial_{u+1}^t(\mathcal{H}_{v+1}^t(k - \binom{t}{v}))|, & \textit{otherwise.} \end{cases}$$

Proof. See Appendix 3.11.2. □

When estimating N_j in the j^{th} round of the algorithm presented in Section 3.4.3, the algorithm has already discovered a collection of cross-layer cuts with size i for each $i < j$, either by sampling or exhaustive enumeration. Let $\bar{N}_i$ be the number of cross-layer cuts with size i seen by the algorithm. Then N_j is lower bounded by $\max\limits_{1 \leq i < j} |\partial_j^m(\mathcal{H}_i^m(\bar{N}_i))|$, where each term $|\partial_j^m(\mathcal{H}_i^m(\bar{N}_i))|$ can be computed easily using the recursive formula in Theorem 3.10. Notice that the original lower bound in Theorem 3.4 is a special case where a single MCLC is assumed and (according to Theorem 3.10) N_j is lower bounded by $|\partial_j^m(\mathcal{H}_d^m(\bar{N}_d = 1))| = \binom{m-d}{j-d}$ for each $j > d$. Theorem 3.10 improves this bound by accounting for more cross-layer cuts, and therefore, it can be used to further reduce the number of iterations required by the Monte Carlo algorithm. We note however that the enhanced lower bounds obtained by Theorems 3.7 and 3.10 may still result in the same order of $O(m^d \log m)$ iterations. Nevertheless, simulation studies in Section 3.6 show that these enhanced bounds can substantially reduce the number of iterations.

Finally, a probabilistic lower bound for N_j can also be established by using the estimated value $\hat{N}_i$ instead of $\bar{N}_i$. In that case, the parameters δ and ϵ need to be adjusted in a way similar to Theorem 3.8.

3.6 Empirical Studies

We present some empirical results about the reliability estimation algorithms. We compare the different lower bounds for N_i produced by the methods described in Sections 3.4 and 3.5, and look at the number of iterations required for different variants of the estimation algorithm. In addition, we will compare the actual accuracy of the failure polynomials computed by the algorithm with the theoretical guarantee provided by the Estimator Theorem.

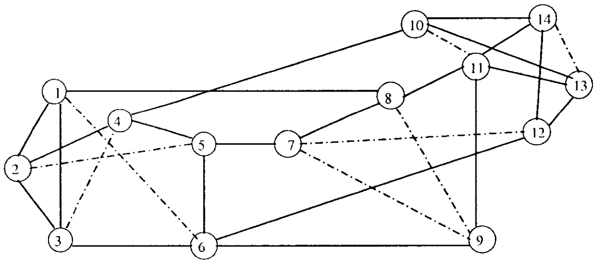


Figure 3-3: The augmented NSFNET.

We used the augmented NSFNET (Figure 3-3) as the physical topology. We generated 350 random logical topologies with 6 to 12 nodes and created lightpath routings using the MCF (Multi-Commodity Flow) algorithms described in Chapter 2. For each lightpath routing, we ran four different reliability estimation algorithms to compute their failure polynomials:

1. ENUM: Each value of N_i is computed by enumeration.

2. MIXED$_{\text{original}}$: The original algorithm that combines the enumeration and Monte Carlo methods, introduced in Section 3.4.3, with $\epsilon = \delta = 0.01$.

3. MIXED$_{\text{KK}}$: The algorithm that combines the enumeration and Monte Carlo methods, using Theorem 3.10 to derive the lower bound for ρ_i.

4. MIXED$_{\text{sample}}$: The algorithm that combines the enumeration and Monte Carlo methods, using the importance sampling technique in Section 3.5 to derive the lower bound for ρ_i. In this case, we have picked $\epsilon_{mc} = 0.01, \epsilon_{lb} = 0.1, \delta_{mc} = \delta_{lb} = \frac{0.005}{30}$. For the collection $\mathcal{C}$ of cross-layer cuts, we only keep the 100 smallest cross-layer cuts.

Table 3.1 shows the average number of iterations required for each algorithm to compute the failure polynomial. The result shows that the combined enumeration and Monte Carlo approach helps to significantly reduce the number of iterations. In addition, the algorithms MIXED$_{\text{KK}}$ and MIXED$_{\text{sample}}$ is able to further reduce the number of iterations by exploiting the knowledge of the discovered cross-layer cuts.

Between the two enhanced algorithms, algorithm MIXED$_{\text{sample}}$ in general achieves a better lower bound, as shown in Figure 3-4, because of the the additional importance sampling step. However, for small regimes of i where the number of iterations dominates, the lower bounds from the two algorithms are close enough that the difference in the number of iterations is small. In addition, since algorithm MIXED$_{\text{sample}}$ requires the additional importance sampling step, the overall number of iterations required by the two algorithms are close to each other.

Algorithm	Monte Carlo Iterations		
	N_i Estimation	$\hat{\rho}_i$ Estimation	Total
ENUM	536,870,912	N/A	536,870,912
MIXED$_{\text{original}}$	46,900,857	N/A	46,900,857
MIXED$_{\text{KK}}$	15,467,815	N/A	15,467,815
MIXED$_{\text{sample}}$	11,968,535	2,485,477	14,454,012

Table 3.1: Number of iterations for each algorithm.

Finally, we compare the actual accuracy of the failure polynomial generated by algorithm MIXED$_{\text{sample}}$ with the theoretical guarantee given by the Estimator Theo-

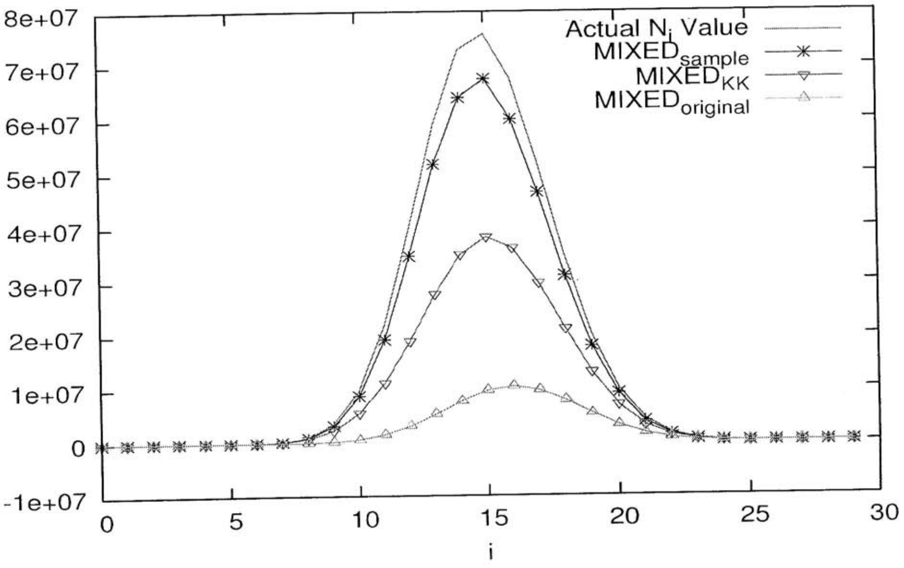


Figure 3-4: Lower bounds for N_i produced by $\text{MIXED}_{\text{original}}$ and $\text{MIXED}_{\text{enhanced}}$.

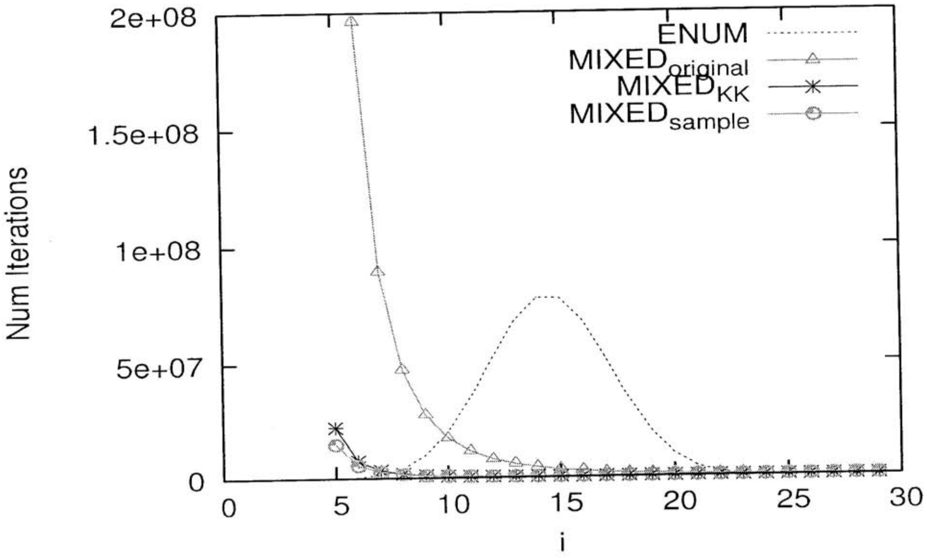


Figure 3-5: Number of iterations to estimate N_i by each algorithm.

rem. Figure 3-6 shows the accuracy results on two sets of failure polynomials, with Monte Carlo parameters $\epsilon = 0.01$ and 0.05. For each set of failure polynomials, we compute the maximum relative error among them for various values of p. Therefore, each curve shows the upper envelope of relative errors by the failure polynomials. In both cases, the relative error is much smaller than the theoretical guarantee. This is because by using a lower bound for ρ_i, the algorithm over-samples in each Monte

Carlo approximation for N_i. In addition, the errors for the N_i estimates are independent and may cancel out each other. Therefore, in practice, the algorithm would provide much better estimates than theoretically guaranteed.

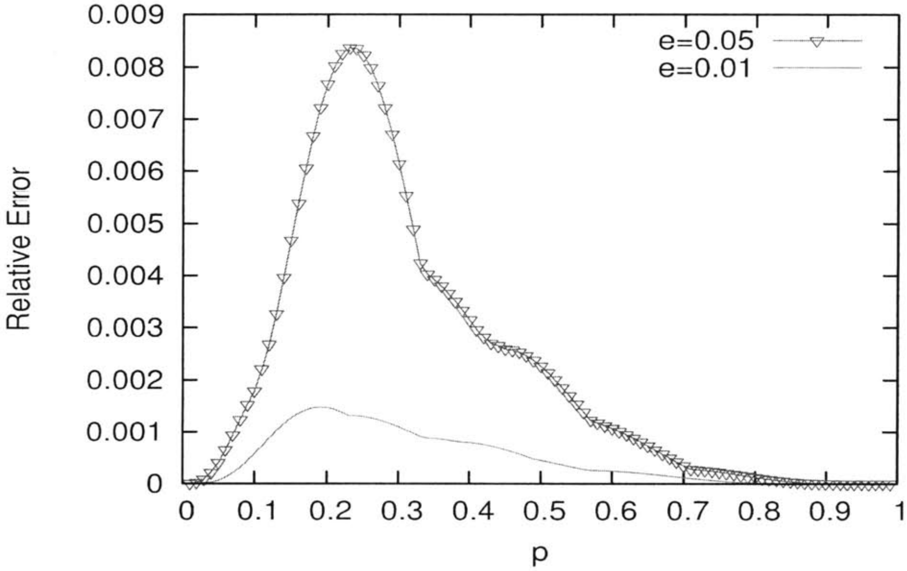


Figure 3-6: Relative error of the failure polynomial approximation.

3.7 Estimating Cross-Layer Reliability with Absolute Error

We have considered computing *relative* approximation for the failure polynomial $F(p)$. However, in certain contexts, it may make sense to describe the error in absolute terms. A function $\hat{F}(p)$ is ϵ-absolute-approximate to $F(p)$ if:

$$|\hat{F}(p) - F(p)| \leq \epsilon.$$

For example, if our goal is to design a network with a certain reliability target (say five 9s), it is sufficient to present a network whose associated failure polynomial has absolute error in the order of 10^{-6}. Constructing a failure polynomial with such relative error, however, may be overly stringent.

A function that is ϵ-approximate to $F(p)$ immediately implies that it is ϵ-absolute-approximate. As it turns out, using a similar approach of probabilistically estimating each N_i requires a much smaller number of samples to achieve ϵ-absolute accuracy. The total number of iterations required to compute an ϵ-absolute-approximation for $F(p)$ with high probability is $O(m \log m)$, in contrast to $O(m^d \log m)$ in the case of ϵ-approximation.

The intuition behind the difference is that, computing an ϵ-approximation for N_i is difficult when the density ρ_i is small. However, in that case, the absolute contribution of the term $N_i p^i (1-p)^{m-i} = \rho_i \binom{m}{i} p^i (1-p)^{m-i}$ will be small as well. Therefore, in this case, even a large relative error for N_i will only account for a small absolute error.

More precisely, by the Estimator Theorem, the Monte Carlo method yields an $(\frac{\epsilon}{\sqrt{\rho_i}}, \frac{\delta}{m+1})$-approximation for N_i with $\frac{4}{\epsilon^2} \ln \frac{2(m+1)}{\delta}$ samples. In other words, if we run the Monte Carlo method with $O(\log m)$ samples to estimate each N_i, we can obtain $\frac{\epsilon}{\sqrt{\rho_i}}$-approximations $\hat{N}_i$ for all N_i with probability at least $1 - \delta$. This implies:

$$
\begin{aligned}
|\sum_{i=0}^{m} (\hat{N}_i - N_i) p^i (1-p)^{m-i}| &\leq \sum_{i=0}^{m} \frac{\epsilon}{\sqrt{\rho_i}} N_i p^i (1-p)^{m-i} \\
&\leq \epsilon \sum_{i=0}^{m} \binom{m}{i} p^i (1-p)^{m-i} = \epsilon.
\end{aligned}
$$

This means that we can compute ϵ-absolute-approximation for the failure polynomial $F(p)$ with high probability with a total of $O(m \log m)$ iterations. Unlike the case for ϵ-approximation, the number of iterations is independent of the Min Cross Layer Cut value d. This makes the method efficient even in the settings where d can be large.

3.8 Improving Reliability via MCLC Maximization

As illustrated in Section 3.1, lightpath routing in a layered network plays an important role in the reliability. Designing a lightpath routing that maximizes reliability, however, is a very complex problem. As we have seen in Figure 3-1, a lightpath routing that is optimal for a certain value of p may not perform as well for other values of

p. This makes the network design aspect of cross-layer reliability a challenging and interesting problem.

In this section, we study the reliability performance of several lightpath routing algorithms presented in Chapter 2, whose objective is to maximize the Min Cross Layer Cut (MCLC). As discussed in Section 3.3.1, maximizing the MCLC is closely related to maximizing reliability, especially for small values of p. The relationship between the two quantities is described by Theorem 3.11. We state the main result relevant to this chapter here. The proof will be given in Chapter 4, where a generalized version of the theorem is presented.

Assume that logical and physical topologies are given. Consider two lightpath routings 1 and 2 for these topologies. Let d be the MCLC of lightpath routing 1, and $F_1(p)$ be its failure polynomial. Similarly, let c and $F_2(p)$ be the MCLC and failure polynomial of lightpath routing 2, respectively. The failure polynomials $F_1(p)$ and $F_2(p)$ are given by

$$F_1(p) = \textstyle\sum_{i=d}^{m} N_i p^i (1-p)^{m-i}$$
$$F_2(p) = \textstyle\sum_{i=c}^{m} M_i p^i (1-p)^{m-i}.$$

Theorem 3.11 *Assume $d > c$. Then, there exists a positive number p_0 such that $F_1(p) < F_2(p)$ for $p < p_0$. In particular,*

$$p_0 = \frac{(c+1)M_C}{2m\binom{m}{c}}.$$

Motivated by Theorem 3.11, we will investigate the reliability performance of the lightpath routing algorithms studied in Chapter 2, whose objectives are to maximize the MCLC.

3.8.1 Simulation Studies

In Chapter 2, we showed that the multi-commodity flow algorithm, $\mathsf{MCF}_{\mathsf{MinCut}}$, and its enhanced version, $\mathsf{MCF}_{\mathsf{LF}}$, outperform the existing survivable lightpath routing algorithm, SURVIVE [76], in terms of MCLC performance. Since MCLC is closely tied to cross-layer reliability, it is therefore interesting to see whether a similar observation holds in terms of reliability to random failures.

We used the augmented NSFNET (Figure 3-3) as the physical topology, and generated 350 random logical topologies with size from 6 to 12 nodes and connectivity at least 4. We study the reliability performance of the three lightpath routing algorithms: $\mathsf{MCF}_{\mathsf{LF}}, \mathsf{MCF}_{\mathsf{MinCut}}$ and SURVIVE. For each lightpath routing generated by the algorithms, we compute an approximate failure polynomial using the technique proposed in Section 3.5, and evaluate its reliability.

Figure 3-7 shows the cumulative distributions of reliability for the lightpath routings generated by the three algorithms, with $p = 0.1$. The multi-commodity flow based algorithms, which try to maximize the MCLC of the lightpath routings, were able to generate more lightpath routings with higher reliability than SURVIVE, whose objective is to find a lightpath routing with MCLC at least two. For small p, the term $N_d p^d (1-p)^{m-d}$, where d is the Min Cross Layer Cut, dominates other terms in the failure polynomial. Therefore, maximizing d has the effect of maximizing the reliability of the network.

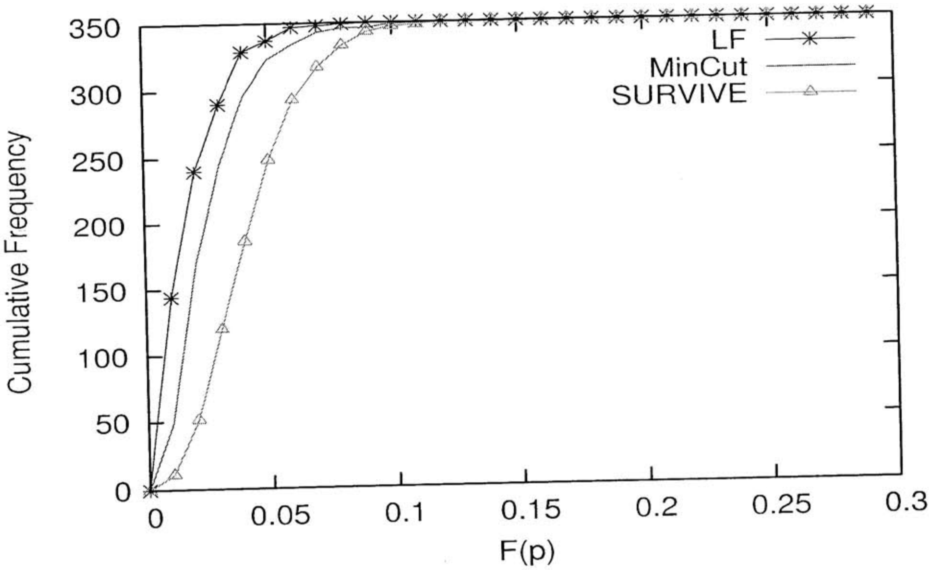
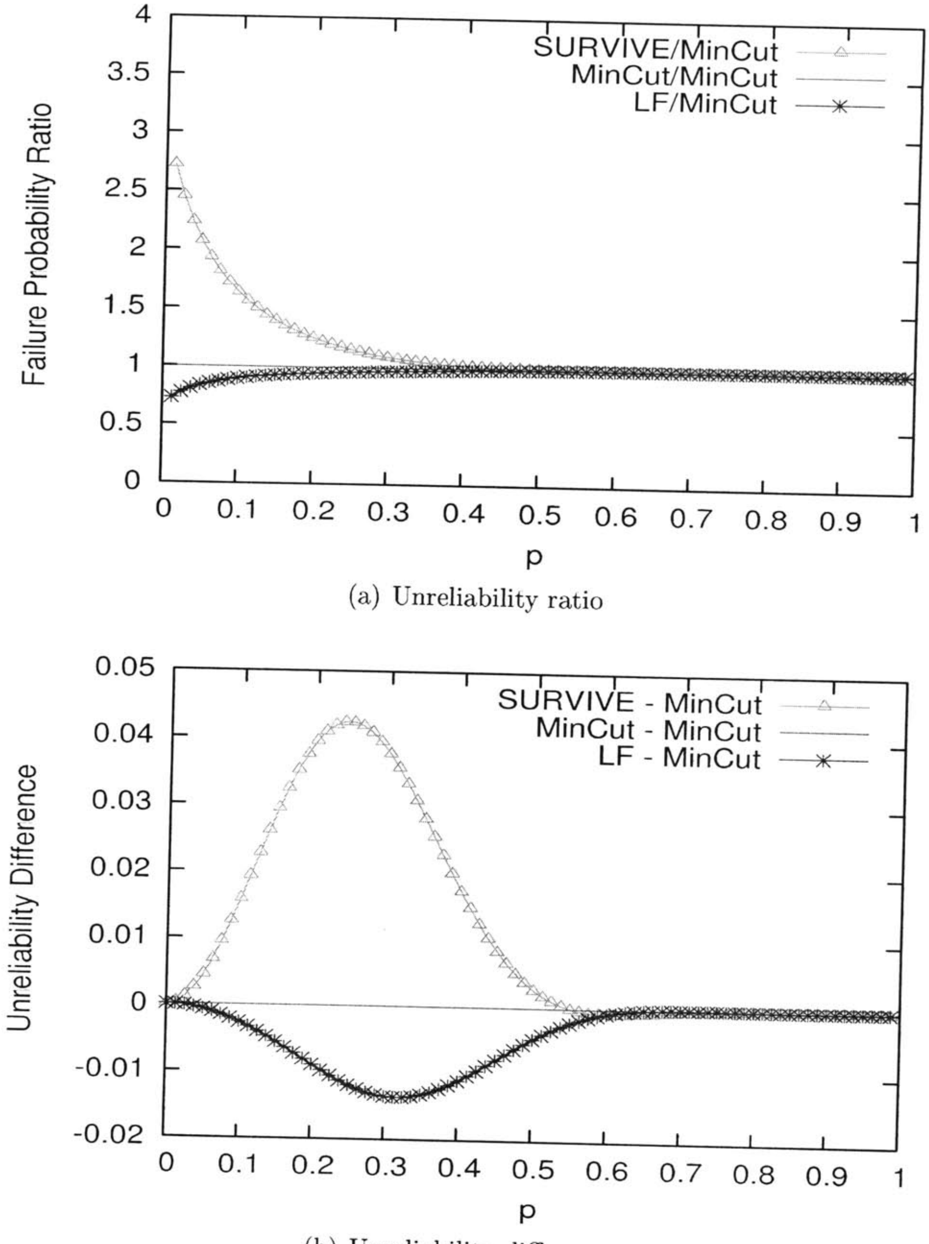


Figure 3-7: Reliability CDF for different algorithms with $p = 0.1$, which shows the number of instances with unreliability less than the value given by the x-axis.

The dependence of reliability on lightpath routing and link failure probability p is further illustrated by Figure 3-8, which plots the ratio and absolute difference of av-



(a) Unreliability ratio

(b) Unreliability difference

Figure 3-8: Ratio and absolute difference of average unreliabilities among different algorithms.

erage failure probabilities of the lightpath routings generated by the three algorithms, using $\mathsf{MCF}_{\mathsf{MinCut}}$ as the baseline. When p is small, the multi-commodity flow routing algorithms are clearly better than SURVIVE in terms of the average reliability. However, as p gets larger, the difference in reliability performance among the algorithms diminishes. In fact, as seen in Figure 3-8(b), the reliability of all three algorithms are very close. This is because for large p, the unreliability for any lightpath routing would be very close to 1.

Figure 3-9 compares the average N_i values of the lightpath routings generated by the algorithms. Again using $\mathsf{MCF}_{\mathsf{MinCut}}$ as the baseline, Figure 3-9 shows that none of the algorithms dominate the others in all N_i values. The multi-commodity flow algorithms try to maximize the Min Cross Layer Cut at the expense of creating more cross-layer cuts of larger size. The objective for SURVIVE, on the other hand, is to minimize the total number of physical hops subject to the constraint that MCLC is at least two. In an environment where p is high, minimizing the physical hops may be a better strategy, as we have seen in Figure 3-1. This is reflected by the fact that lightpaths routings produced by SURVIVE have smaller average N_i values when i is large.

In the setting of WDM networks, we expect p to be typically small. Therefore, maximizing the Min Cross Layer Cut appears to be a reasonable strategy. However, it is important to keep in mind that the same insight may not apply to other settings where physical links fail with high probability (e.g. Delay Tolerant Networks).

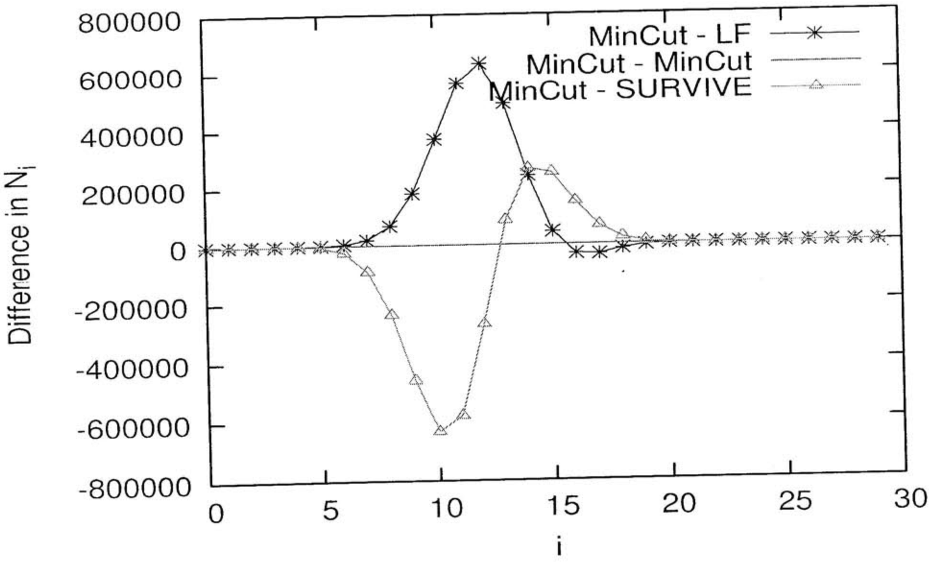


Figure 3-9: Difference in average N_i values among different algorithms.

3.9 Extensions to the Failure Model

In this section, We present a few extensions to the failure model and discuss the application of the reliability estimation method to these extensions.

3.9.1 Non-uniform Failure Probabilities

In the non-uniform physical link failure model, each physical link (i,j) fails with probability p_{ij}. The physical topology can be approximated by replacing each physical link (i,j) by $k = round\left(\frac{\log(1-p_{ij})}{\log(1-p')}\right)$ physical links in series, where $round()$ is the rounding function and p' is a constant that represents the link failure probability of the transformed network (Figure 3-10). In this case, the probability that none of the replacements for (i,j) fail equals:

$$(1-p')^k = (1-p')^{\frac{\log(1-p_{ij})}{\log(1-p')}}(1-p')^{\epsilon} = (1-p_{ij})(1-p')^{\epsilon},$$

where $|\epsilon| = |round\left(\frac{\log(1-p_{ij})}{\log(1-p')}\right) - \left(\frac{\log(1-p_{ij})}{\log(1-p')}\right)| \le 0.5$. Therefore, this probability can be made arbitrarily close to $1-p_{ij}$ by choosing a sufficiently small p', with the tradeoff being a larger number of new links. In this case, the lightpath routing can then be modified such that a logical link originally using (i,j) is now routed over its replacements. This gives us an equivalent layered network where every physical link fails independently with probability p'.

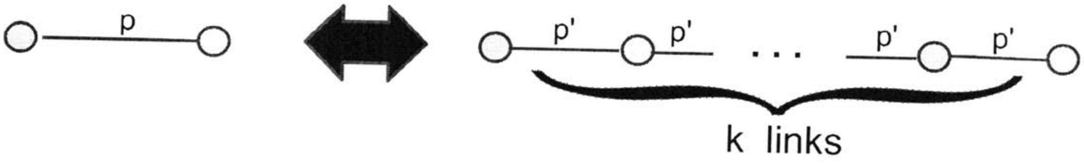


Figure 3-10: A physical link with failure probability p is equivalent to $k = \log(1-p)/\log(1-p')$ physical links in series with failure probability p'.

3.9.2 Random Node Failures

The reliability estimation method can be extended to a model where each physical link fails with probability p and each physical node fails with probability q. We can model a network state as the set of failed physical nodes and links, and a logical link will fail if any of the physical nodes and links it uses fail. In this case, a cross-layer cut is a set of physical nodes and links whose failures would cause the logical topology to be disconnected. The reliability of the layered network can then be expressed as follows:

$$\sum_{i=0}^{m}\sum_{j=0}^{n} N_{ij}p^i(1-p)^{m-i}q^j(1-q)^{n-j},$$

where m, n are the numbers of physical links and nodes respectively, and N_{ij} is the number of cross-layer cuts with i failed physical links and j failed physical nodes. Then we can estimate the reliability in a similar fashion, by approximating each N_{ij} separately via the Monte-Carlo method. To estimate N_{ij}, network states with i fibers and j nodes will be uniformly sampled. The methods in Sections 3.4.2 and 3.5.1 to establish lower bounds on N_i can be extended to establish lower bounds on N_{ij}, based on a similar observation in this setting that any network state that contains a cross-layer cut is also a cross-layer cut.

3.10 Conclusion

We consider network reliability in multi-layer networks. In this setting, logical link failures can be correlated even if physical links fail independently. Hence, conventional estimation methods that assume particular topologies, independent failures, and network parameters cannot be used for our problem. To that end, we develop a Monte Carlo simulation based estimation algorithm that approximates cross-layer reliability with high probability. We first extend the classical polynomial expression for reliability to multi-layer networks. Our algorithm approximates the failure polynomial by estimating the values of its coefficients. The advantages of our approach are two fold. First, it does not require resampling for different values of link failure probability p. Second, with a polynomial number of iterations, it guarantees the accuracy of estimation with high probability. We also observe through the polynomial expression that lightpath routings that maximize the MCLC can perform very well in terms of reliability. This observation leads to the development of lightpath routing algorithms that attempt to maximize reliability.

While sampling failure states, our estimation algorithm naturally reveals the vulnerable parts of the network or lightpath routing. This information can be used to enhance the current lightpath routing. In Chapter 5 we will explore different approaches of improving the reliability of a network using such information.

3.11 Chapter Appendix

3.11.1 Approximating Union of Sets

As seen in Section 3.5, given a set of cross-layer cuts $\mathcal{F}$, the value $\frac{|\partial_i(\mathcal{F})|}{\binom{m}{i}}$ gives a lower bound for ρ_i. We will discuss in this section how to estimate the size of $\partial_i(\mathcal{F}) = \cup_{C_j \in \mathcal{F}} \partial_i(C_j)$ probabilistically.

Computing the value of $|\partial_i(\mathcal{F})|$ can be formulated as the Union of Sets Problem [64], where Monte Carlo method exists to estimate the size of $|\partial_i(\mathcal{F})|$ using the technique of importance sampling. Here, we define the ground set U to be:

$$U := \{(S, j) : j \in \{1, \ldots, |\mathcal{F}|\}, S \in \partial_i(C_j)\},$$

and the events of interest G to be

$$G := \{(S, j) : S \in \partial_i(\mathcal{F}), j = \min\{k : S \in \partial_i(C_k)\}\}.$$

In other words, the ground set U represents a multi-set where each set S in $\partial_i(\mathcal{F})$ is represented k times in U, where k is the number of elements in $\mathcal{F}$ that are subsets of S. On the other hand, each set S in $\partial_i(\mathcal{F})$ is represented by exactly one element (S, j) in G, where C_j is the first element in $\mathcal{F}$ that is a subset of S. As a result, for each $S \in \partial_i(\mathcal{F})$, $|\{(T, j) \in U : T = S\}| \leq |\mathcal{F}|$, and $|\{(T, j) \in G : T = S\}| = 1$. It immediately follows that:

$$|G| = |\partial_i(\mathcal{F})|,$$

and

$$\frac{|G|}{|U|} \geq \frac{1}{|\mathcal{F}|}.$$

Therefore, by the Estimator Theorem, if we sample from U uniformly at random for T times, where:

$$T = \frac{4|\mathcal{F}|}{\epsilon_{lb}^2} \ln \frac{2}{\delta_{lb}} \geq \frac{4}{\epsilon_{lb}^2 \frac{|G|}{|U|}} \ln \frac{2}{\delta_{lb}},$$

the Monte Carlo method will yield an ϵ_{lb}-approximation for $|G|$, which is equal to $|\partial_i(\mathcal{F})|$, with probability at least $1 - \delta_{lb}$.

Finally, the sample space U can be sampled uniformly at random as follows:

1. Select an element j from $\{1, \ldots, |\mathcal{F}|\}$, where the probability of selecting j is $\frac{|\partial_i(C_j)|}{\sum_{C_k \in \mathcal{F}} |\partial_i(C_k)|}$. Note that $|\partial_i(C_j)| = \binom{m-|C_j|}{i-|C_j|}$, which can be computed easily.

2. Given the selected value j, pick a set $S \in \partial_i(C_j)$ uniformly at random.

The probability of selecting each element $(S, j) \in U$ is therefore:

$$\frac{|\partial_i(C_j)|}{\sum\limits_{C_k \in \mathcal{F}} |\partial_i(C_k)|} \cdot \frac{1}{|\partial_i(C_j)|} = \frac{1}{\sum\limits_{C_k \in \mathcal{F}} |\partial_k(C_k)|} = \frac{1}{|U|}.$$

This gives us a method to establish a probabilistic lower bound $\hat{\rho}_i$ for ρ_i.

3.11.2 Proof of Theorem 3.10

Let $[m] = \{1, \ldots, m\}$ and let $\mathcal{H}_i^m = \{S \subseteq [m] : |S| = i\}$ be a family of subsets of $[m]$ with size i, and let $\mathcal{H}_i^m(k)$ be the first k subsets in $\mathcal{H}_i^m$ under the lexicographical ordering. In addition, for any family $\mathcal{F}$ of subsets of $[m]$ and for any $j > i$, let $\partial_j^m(\mathcal{F})$ be the j^{th} upper shadow of $\mathcal{F}$ over $[m]$. Theorem 3.10 states that:

Theorem 3.10 *For $i < j \leq m$ and $1 \leq k \leq \binom{m}{i}$, let $w = \max\{0 \leq r < i : \binom{m-r}{i-r} \geq k\}$. Also, let $t = m - (w+1), u = j - (w+1)$ and $v = i - (w+1)$. Then:*

$$|\partial_j^m(\mathcal{H}_i^m(k))| = \begin{cases} \binom{m-i}{j-i}, & \textit{if } k = 1 \\ \binom{t}{u} + |\partial_{u+1}^t(\mathcal{H}_{v+1}^t(k - \binom{t}{v}))|, & \textit{otherwise.} \end{cases}$$

The case for $k = 1$ follows from the fact that for a set with size i, it has $\binom{m-i}{j-i}$ supersets with size j. We will prove the case where $k > 1$ in the rest of the section.

Let S be the lexicographically largest element in $\mathcal{H}_i^m(k)$. We first prove the following lemma:

Lemma 3.12 $[w] \subseteq S$ *and* $w + 1 \notin S$.

Proof. Suppose the lemma is not true. We have the following two cases:

1. S does not contain some element $e \in [w]$. In this case, all subsets of $\mathcal{H}_i^m$ that contains $[w]$ are lexicographically smaller than S and thus belong to $\mathcal{H}_i^m(k)$. Therefore, $k = |\mathcal{H}_i^m(k)| > \binom{m-w}{i-w}$. This contradicts with the fact that $\binom{m-w}{i-w} \geq k$.

2. S contains $[w+1]$. So any set $T \in \mathcal{H}_i^m$ that does not contain $[w+1]$ is lexicographically greater than S, and therefore cannot be in $\mathcal{H}_i^m(k)$. As a result, $k = |\mathcal{H}_i^m(k)| \leq \binom{m-(w+1)}{i-(w+1)}$. However, by definition of w, we have $\binom{m-(w+1)}{i-(w+1)} < k$, which is a contradiction.

□

Corollary 3.13 *All elements in $\mathcal{H}_i^m(k)$ must contain $[w]$.*

Proof. Any element in $\mathcal{H}_i^m(k)$ must be lexicographically at most S, and therefore must contain $[w]$. □

Corollary 3.14 *All elements in $\mathcal{H}_i^m$ that contain $[w+1]$ are in $\mathcal{H}_i^m(k)$.*

Proof. Any element that contains $[w+1]$ are lexicographically smaller than S, and therefore belongs to $\mathcal{H}_i^m(k)$. □

We now partition the family $\mathcal{H}_i^m(k)$ into two sub-families:

- $\mathcal{H}_i^m(k)^+ := \{T \in \mathcal{H}_i^m(k) : w+1 \in T\}$
- $\mathcal{H}_i^m(k)^- := \{T \in \mathcal{H}_i^m(k) : w+1 \notin T\}$

As a result of Corollaries 3.13 and 3.14, $\mathcal{H}_i^m(k)^+$ consists of all $\binom{m-(w+1)}{i-(w+1)}$ elements in $\mathcal{H}_i^m$ that contain $[w+1]$, and $\mathcal{H}_i^m(k)^-$ consists of the next $k - \binom{m-(w+1)}{i-(w+1)}$ elements in the lexicographical order. We define a bijection g_w on $\mathcal{H}_i^m(k)^-$ as follows:

$$g_w(T) := \{e - (w+1) : e \in T - [w]\}\,,\ \forall T \in \mathcal{H}_i^m(k)^-. \tag{3.6}$$

In other words, for any $T \in \mathcal{H}_i^m(k)^-$, we construct $g_w(T)$ by first removing the common subset $[w]$ from T and then subtracting each remaining element by $w+1$. As a result, each $g_w(T)$ is a subset of $[m-(w+1)]$. The image $g_w(\mathcal{H}_i^m(k)^-)$ consists of the first $k - \binom{m-(w+1)}{i-(w+1)}$ subsets of $[m-(w+1)]$ size $i-w$ in lexicographical order. In other words, we have:

$$g_w(\mathcal{H}_i^m(k)^-) = \mathcal{H}_{i-w}^{m-(w+1)}(k - \binom{m-(w+1)}{i-(w+1)}). \tag{3.7}$$

Now, consider $\partial_j^m(\mathcal{H}_i^m(k))$, the j^{th} upper shadow over $[m]$ for $\mathcal{H}_i^m(k)$. As a result of Corollary 3.13, all elements in $\partial_j^m(\mathcal{H}_i^m(k))$ must contain $[w]$. We can therefore partition $\partial_j^m(\mathcal{H}_i^m(k))$ in a similar fashion:

- $\partial_j^m(\mathcal{H}_i^m(k))^+ := \{T \in \partial_j^m(\mathcal{H}_i^m(k)) : w+1 \in T\}$
- $\partial_j^m(\mathcal{H}_i^m(k))^- := \{T \in \partial_j^m(\mathcal{H}_i^m(k)) : w+1 \notin T\}$

We now prove the following properties of $\partial_j^m(\mathcal{H}_i^m(k))^+$ and $\partial_j^m(\mathcal{H}_i^m(k))^-$, which allow us to express the cardinality of the upper shadow in Theorem 3.10.

Lemma 3.15 $\partial_j^m(\mathcal{H}_i^m(k))^+ = \{T \in \mathcal{H}_j^m : [w+1] \subset T\}$.

Proof. Every element T in $\partial_j^m(\mathcal{H}_i^m(k))^+$ must contain $[w]$, by Corollary 3.13, and $w+1$, by definition. Therefore, T must contain $[w+1]$. In addition, for any element

T in $\mathcal{H}_j^m$ that contains $[w+1]$, let U be the set with the i smallest elements in T. Since $i \geq w+1$, U contains $[w+1]$ and is in $\mathcal{H}_i^m(k)$ by Corollary 3.14. As a result, the subset T, being a superset of U, is in the j^{th} upper shadow of $\mathcal{H}_i^m(k)$. □

Corollary 3.16 $|\partial_j^m(\mathcal{H}_i^m(k))^+| = \binom{m-(w+1)}{j-(w+1)}$.

Lemma 3.17

$$g_w(\partial_j^m(\mathcal{H}_i^m(k))^-) = \partial_{j-w}^{m-(w+1)}(g_w(\mathcal{H}_i^m(k)^-)).$$

Proof. For any element $T \in \partial_j^m(\mathcal{H}_i^m(k))^-$, there must exist an element $U \in \mathcal{H}_i^m(k)$ such that $U \subset T$. Since $w+1 \notin T$, it follows that $w+1 \notin U$, which implies $U \in \mathcal{H}_i^m(k)^-$. By applying the same bijection g_w to $\partial_j^m(\mathcal{H}_i^m(k))^-$, $g_w(T)$ is a subset of $[m-(w+1)]$ with size $j-w$, and is a superset of $g_w(U)$. In other words:

$$g_w(\partial_j^m(\mathcal{H}_i^m(k))^-) \subseteq \partial_{j-w}^{m-(w+1)}(g_w(\mathcal{H}_i^m(k)^-)).$$

Now given $T \in \partial_{j-w}^{m-(w+1)}(g_w(\mathcal{H}_i^m(k)^-))$, there exists $U \in g_w(\mathcal{H}_i^m(k)^-)$ such that $U \subset T$. It follows that $g_w^{-1}(U) \subset g_w^{-1}(T)$. Since $g_w^{-1}(U) \in \mathcal{H}_i^m(k)^-$, it follows that $g_w^{-1}(T) \in \partial_j^m(\mathcal{H}_i^m(k)^-)$. Therefore, $T \in g_w(\partial_j^m(\mathcal{H}_i^m(k)^-))$, which means

$$g_w(\partial_j^m(\mathcal{H}_i^m(k))^-) \supseteq \partial_{j-w}^{m-(w+1)}(g_w(\mathcal{H}_i^m(k)^-)),$$

which proves the lemma. □

Corollary 3.18

$$|\partial_j^m(\mathcal{H}_i^m(k))^-| = |\partial_{j-w}^{m-(w+1)}(\mathcal{H}_{i-w}^{m-(w+1)}(k - \binom{m-(w+1)}{i-(w+1)}))|.$$

Proof.

$$\begin{aligned}
|\partial_j^m(\mathcal{H}_i^m(k))^-| &= |g_w(\partial_j^m(\mathcal{H}_i^m(k))^-)| \\
&= |\partial_{j-w}^{m-(w+1)}(g_w(\mathcal{H}_i^m(k)^-))| \\
&= |\partial_{j-w}^{m-(w+1)}(\mathcal{H}_{i-w}^{m-(w+1)}(k - \binom{m-(w+1)}{i-(w+1)}))|.
\end{aligned}$$

The second equality is due to Lemma 3.17, and the third equality is due to Equation (3.7). □

The expression for $|\partial_j^m(\mathcal{H}_i^m(k))|$ for $k > 1$ follows immediately from Corollaries 3.16 and 3.18.

3.11.3 Estimating Reliability by Importance Sampling

As discussed in Section 3.4, estimating reliability by directly simulating physical link failures requires a large sample size when the link failure probability p is small, due to the large coefficient of variation of the estimator. In this section, we discuss how importance sampling can be used to reduce the coefficient of variation.

Given the physical, logical topologies and a lightpath routing. Let $\mathcal{H}$ be the sample space, that is, all possible subsets of the physical links E_P. Given a network state $S \in \mathcal{H}$, the 0-1 random variable $U(S)$ is defined to be 1 if and only if S is a cross-layer cut. Suppose each physical link fails with probability p. Then the unreliability of the layered network is simply the expected value, $E_p(U)$, of U, where the subscript p indicates that the expectation is taken over the probability distribution where every physical link fails with probability p. It can be written as follows:

$$\begin{aligned} E_p(U) &= \sum_{S \in \mathcal{H}} U(S) Pr(S) \\ &= \sum_{i=0}^{m} N_i p^i (1-p)^{m-i} \\ &= \sum_{i=0}^{m} N_i \frac{p^i (1-p)^{m-i}}{p'^i (1-p')^{m-i}} p'^i (1-p')^{m-i} \\ &= E_{p'}(U'), \end{aligned} \tag{3.8}$$

where N_i is the number of cross-layer cuts with size i, and U', called the *likelihood ratio estimator*, is a random variable on $\mathcal{H}$ such that:

$$U'(S) = \begin{cases} \frac{p^i(1-p)^{m-i}}{p'^i(1-p')^{m-i}}, & \text{if } S \text{ is a cross-layer cut, where } |S| = i \\ 0, & \text{if } S \text{ is not a cross-layer cut.} \end{cases}$$

Equation (3.8) implies that the expected value for U at link failure probability p is equal to the expected value for U' at link failure probability p'. Therefore, we can sample the value for U' at link failure probability p' to obtain an unbiased estimate on the unreliability of the network at link failure probability p. The variance of U' is given by:

$$\begin{aligned} Var_{p'}(U') &= E_{p'}(U'^2) - \Big(E_{p'}(U')\Big)^2 \\ &= \sum_{i=0}^{m} N_i p'^i (1-p')^{m-i} \left(\frac{p^i(1-p)^{m-i}}{p'^i(1-p')^{m-i}}\right)^2 - (E_p(U))^2 \\ &= \sum_{i=0}^{m} N_i p^i (1-p)^{m-i} \left(\frac{p^i(1-p)^{m-i}}{p'^i(1-p')^{m-i}}\right) - (E_p(U))^2 \\ &= E_p(U') - (E_p(U))^2 . \end{aligned} \tag{3.9}$$

Estimating Unreliability at Small p With Importance Sampling

The major design decision involved in importance sampling is the choice of the new sampling distribution, which, in our case, is the choice of p'. Since direct sampling is less effective when the link failure probability p is small, it makes sense to choose p' to optimize for this case.

Consider a lightpath routing with Min Cross Layer Cut value d. When p is sufficiently small, the value of $E_p(U)$ can be bounded as follows:

$$
\begin{aligned}
N_d p^d (1-p)^{m-d} &\leq \sum_{i=d}^{m} N_i p^i (1-p)^{m-i} && (3.10)\\
&= N_d p^d (1-p)^{m-d} + \sum_{i=d+1}^{m} N_i p^i (1-p)^{m-i} \\
&\leq (1+\epsilon') N_d p^d (1-p)^{m-d}, && (3.11)
\end{aligned}
$$

where ϵ' is a small constant. Similarly, the value of $E_p(U')$ can be bounded as follows:

$$
\begin{aligned}
N_d \frac{p^{2d}(1-p)^{2(m-d)}}{p'^d(1-p')^{m-d}} &\leq \sum_{i=d}^{m} N_i p^i (1-p)^{m-i} \left(\frac{p^i(1-p)^{m-i}}{p'^i(1-p')^{m-i}} \right) && (3.12)\\
&= N_d \frac{p^{2d}(1-p)^{2(m-d)}}{p'^d(1-p')^{m-d}} + \sum_{i=d+1}^{m} N_i \frac{p^{2i}(1-p)^{2(m-i)}}{p'^i(1-p')^{m-i}} \\
&\leq (1+\epsilon') N_d \frac{p^{2d}(1-p)^{2(m-d)}}{p'^d(1-p')^{m-d}}; && (3.13)
\end{aligned}
$$

Therefore, the squared coefficient of variation for U' is bounded as follows:

$$\frac{Var_{p'}(U')}{E_{p'}(U')^2} = \frac{E_p(U')}{E_p(U)^2} - 1, \qquad \text{by (3.9)}$$

$$\leq \frac{(1+\epsilon')N_d p^{2d}(1-p)^{2(m-d)}}{p'^d(1-p')^{m-d}} \cdot \frac{1}{N_d^2 p^{2d}(1-p)^{2(m-d)}} - 1,$$

by (3.10) and (3.13)

$$\leq \frac{1+\epsilon'}{N_d p'^d(1-p')^{m-d}} - 1,$$

and:

$$\frac{Var_{p'}(U')}{E_{p'}(U')^2} = \frac{E_p(U')}{E_p(U)^2} - 1, \qquad \text{by (3.9)}$$

$$\geq \frac{N_d p^{2d}(1-p)^{2(m-d)}}{p'^d(1-p')^{m-d}} \cdot \frac{1}{(1+\epsilon')^2 N_d^2 p^{2d}(1-p)^{2(m-d)}} - 1,$$

by (3.11) and (3.12)

$$= \frac{1}{(1+\epsilon')^2 N_d p'^d(1-p')^{m-d}} - 1.$$

The term $\frac{1}{N_d p'^d(1-p')^{m-d}}$ is minimized when $p' = \frac{d}{m}$. Therefore, if the Monte Carlo method samples network states at $p' = \frac{d}{m}$, the squared coefficient of variation will be:

$$\begin{aligned}\frac{Var_{p'}(U')}{E_{p'}(U')^2} &= \Theta\left(\frac{1}{(\frac{d}{m})^d(1-\frac{d}{m})^{m-d}}\right) \\ &= \Theta\left(\left(\frac{m-d}{d}\right)^d (1-\frac{d}{m})^{-m}\right) \\ &= \Theta(m^d).\end{aligned}$$

Therefore, the sample size to establish a (ϵ, δ)-approximation, which is proportional to the squared coefficient of variation [97], is $\Theta(m^d)$ when p is small. Like the algorithm introduced in Section 3.4, the knowledge of the Min Cross Layer Cut value d is needed to carry out the Monte Carlo method efficiently. This gives us the following importance sampling algorithm IS to effciently estimate the cross-layer reliability when p is small.

Algorithm 3 IS

```
Compute MCLC value $d$ for the lightpath routing.
Simulate, for $T = \Theta(m^d)$ times, the event that each phyical link fails with probability $p' = \frac{d}{m}$. Let $\mathcal{C}$ be the set of the $T$ samples collected.
For each $i \in \{0, \ldots, m\}$, count the number of cross-layer cuts in $\mathcal{C}$ with exactly $i$ physical links, and denote the count as $M_i$.
For any link failure probability $p$, the estimated unreliability is given by $\sum_{i=0}^{m} M_i \frac{p^i(1-p)^{m-i}}{p'^i(1-p')^{m-i}}$.
```

By setting p' to $\frac{d}{m}$, the algorithm IS is maximizing the likelihood of sampling network states with d fibers, thereby achieving the best estimate on the number of small cross-layer cuts, which contribute to the majority of the unreliability when p is small.

Compared to this importance sampling approach, the algorithm introduced in Section 3.4 requires a total of $O(m^d \log m)$ samples to estimate all values of N_i. However, the output of the algorithm allows us to estimate the cross-layer reliability accurately for *all* values of p. Note that the majority of the computation is allocated to estimate the values of N_i where i is close to d. In particular, similar to importance sampling, the algorithm requires $O(m^d)$ samples to compute the value of N_d, by enumerating all $O(m^d)$ possible network states with d fibers. In this regard, both algorithms require a similar amount of computation to obtain a good estimate of N_d, in order to accurately estimate the cross-layer reliability when p is small.

In IS, since the value of p' is chosen to optimize for small p , the relative error on the reliability estimate for large p can be large if the same set of samples is used. For instance, when $p = \frac{m-1}{m}$, the variance of the estimator $Var_{p'}(U')$ is given by:

$$\begin{aligned}
Var_{p'}(U') &= E_p(U') - (E_p(U))^2, \quad \text{by Equation (3.9)} \\
&\geq \sum_{i=0}^{m} N_i \left(\frac{p^2}{p'}\right)^i \left(\frac{(1-p)^2}{1-p'}\right)^{m-i} - 1 \\
&= \sum_{i=0}^{m} N_i \left(\frac{((m-1)/m)^2}{(d/m)}\right)^i \left(\frac{(1/m)^2}{1-(d/m)}\right)^{m-i} - 1 \\
&= \sum_{i=0}^{m} N_i \left(\frac{(m-1)^2}{md}\right)^i \left(\frac{1}{m(m-d)}\right)^{m-i} - 1 \\
&\geq N_{m-1} \left(\frac{(m-1)^2}{md}\right)^{m-1} \left(\frac{1}{m(m-d)}\right) - 1 \\
&= \Theta\left(N_{m-1} \left(\frac{m}{d}\right)^{m-3}\right).
\end{aligned}$$

In other words, if the samples are collected with link failure probability $p' = \frac{d}{m}$, the algorithm IS will require at least $\Theta((\frac{m}{d})^{m-3})$ samples in order to approximate the cross-layer reliability at $p = \frac{m-1}{m}$ to a constant relative error. Therefore, to efficiently estimate the cross-layer reliability accurately for all values of p, the algorithm IS needs to be extended to collect samples at various link failure probabilities p'. In that case, the sampling plan will become quite similar to the algorithm in Section 3.4, which explicitly controls the collection of network states with different sizes by sampling network states of each size separately.

Chapter 4

Optimal Reliability Conditions for Lightpath Routings

4.1 Introduction

In the previous chapter, we have defined the *cross-layer reliability* to quantify network survivability under random physical failures; and developed an algorithm to estimate the cross-layer reliability function. This allows us to assess the reliability of a layered network under different link failure probabilities. One important observation we made is that a lightpath routing that is good at one failure probability may not perform as well as other lightpath routings under a different failure probability. As such, optimal lightpath routings under different failure probabilities may have different characteristics.

The goal of this chapter is to study the relationship between the link failure probability, the cross-layer reliability and the structure of a layered network. The understanding of such will shed light on desirable properties for a reliable layered network in different failure probability regimes. The key to our study is the cross-layer failure polynomial introduced in Chapter 3. The coefficients of the polynomial contain the structural information about the cross-layer topology and lightpath routing. The study of the polynomial allows us to formulate the optimality condition and provides important insights on how lightpath routing should be designed for better reliability, which will be the focus of Chapter 5.

This chapter is organized as follows. In Section 4.2 we discuss the previous work on designing reliable single-layer networks under random link failures, and discuss the applicability of these results to our multi-layer models. We will review our network and failure model in Section 4.3, and discuss some concepts that are important to our study in the following sections. In Section 4.4, we identify the conditions for optimal lightpath routings in different failure probability regimes. Namely, in the low probability regime, maximizing the min cut of the (layered) network maximizes reliability, whereas in the high probability regime, minimizing the spanning tree of the network maximizes reliability. The results from Section 4.4 are extended in Section 4.5, in which additional information about the layered network is taken into account in the analysis, which leads to a stronger result that unifies the results in the previous sections. Finally, in Section 4.6, we carry out empirical studies to examine various attributes of lightpath routings optimized for the different failure probability regimes, as well as compare the bounds developed in Section 4.5 with the actual values.

4.2 Related Work

The problem of designing reliable networks has been studied rather extensively in the single-layer setting. In the single-layer network design problem, the goal is to construct the most reliable graph topology, given the number of nodes and the number of edges. An important concept here is that of *uniformly optimally reliable (UOR) graph*; a graph is uniformly optimally reliable if for all the values of link failure probability it yields the best reliability among the graphs using the same numbers of nodes and edges. The work in [21,116] studied the conditions for a UOR graph to exist. However, a UOR graph does not always exist [79], and hence, it is also important to study *locally optimally reliable (LOR) graphs.* In [16], the authors characterized the class of LOR graphs for different failure probability regimes. More details on the class of UOR graphs and LOR graphs can be found in [9,10,20,80].

The reliable network design problem in a layered setting consists of three components: logical topology design, physical topology design, and lightpath routing design. In layered networks, careful design of the physical and logical topologies alone does not immediately translate to high reliability, as the lightpath routing also plays a crucial role. In this chapter, we focus on reliable lightpath routing design assuming that the logical and physical topologies are given. As we will see in the following sections, some of the important insights behind reliable topology design in the single-layer can be adopted to our lightpath routing design problem.

4.3 Failure Polynomial and Connectivity Parameters

We consider the same network and failure model as in Chapter 3, where a layered network consists of the logical topology $G_L = (V_L, E_L)$ built on top of the physical topology $G_P = (V_P, E_P)$ through a lightpath routing. The number of physical links $|E_P|$ is denoted by m, and each physical link fails independently with probability p. When a physical link fails, all logical links that use the physical link also fail. The reliability of the layered network is defined to be the probability that the logical toplogy remains connected.

Recall that the reliability of the lightpath routing can be expressed as the *failure polynomial* (Section 3.3.1):

$$F(p) = \sum_{i=0}^{m} N_i p^i (1-p)^{m-i}. \tag{4.1}$$

Each coefficient N_i represents the number of cross-layer cuts of size i in the network. Define a *Min Cross Layer Cut (MCLC)* as a smallest set of physical links needed to disconnect the logical network. Denote by d the size of MCLC, then d is the smallest i such that $N_i > 0$, meaning that the logical network will not be disconnected by fewer than d physical link failures. As discussed in Chapter 2, the MCLC is a generalization of single-layer min-cut to the multi-layer setting.

Define a *Max Cross Layer Non-Cut (MCLNC)* as a largest set of physical links whose failure would not disconnect the logical network. Denote by c the size of

MCLNC, then c is the maximum number of fiber failures that the logical network can *possibly* survive. Since $N_i \leq \binom{m}{i}$, by definition, c is the largest i such that $N_i < \binom{m}{i}$, and we have $N_i = \binom{m}{i}, \forall i > c$, meaning that more than c failures would always disconnect the logical network.

The Cross Layer Non-Cuts are closely related to the Cross-Layer Spanning Trees, defined in Section 2.2.3 as a *minimal* set of fibers whose survival keeps the logical network connected. Hence, if T is a cross-layer spanning tree, then the survival of just $T \setminus \{(i,j)\}$ renders the logical network disconnected for any fiber $(i,j) \in T$. Note that this is a generalization of the single-layer spanning tree. However, unlike a single-layer graph where all spanning trees have the same size, in a layered graph, spanning trees can have different sizes. Thus, we define a *Min Cross Layer Spanning Tree (MCLST)* as a cross-layer spanning tree with minimum number of physical links.

Each Max Cross Layer Non-Cut corresponds to a Min Cross Layer Spanning Tree, and vice versa. That is, for an MCLNC S, $E_P \setminus S$ is an MCLST because the survival of $E_P \setminus S$ keeps the logical network connected, yet the removal of any additional link would disconnect the network. Consequently, the value $b = m - c$ is the size of Min Cross Layer Spanning Tree (MCLST), and any result with MCLNC directly translates into a result with MCLST, and vice versa. In the following, we will use both terms interchangeably.

Note that for given logical and physical topologies, MCLC and MCLST are all determined by the lightpath routing. Consider again the examples in Figure 3-1. The disjoint routing in Figure 3-1(c), which has better reliability for small p, has $d = 2$ and $b = 3$. On the other hand, the shortest routing in Figure 3-1(d), which has better reliability for large p, has $d = 1$ and $b = 2$. Furthermore, the optimal routing in Figure 3-1(e) has $d = 2$ and $b = 2$. This example suggests that maximizing MCLC may lead to better reliability for small p, while minimizing MCLST may lead to better reliability for large p. It turns out that this is true in general, and this will be further discussed in Section 4.4.

4.4 Properties of Optimal Lightpath Routings

Based on the failure polynomial of a lightpath routing, and its associated connectivity parameters, one can develop inights into optimal lightpath routing under different probability regimes. In Section 3.8 we have mentioned that a lightpath routing with a higher MCLC value will have higher reliability for sufficiently small link failure probability p. In this section, we will discuss in greater details the optimal lightpath routings in different failure probability regimes.

4.4.1 Uniformly and Locally Optimal Lightpath Routings

We start with a discussion of routings that are most reliable for all failure probabilities. The observations in this section will motivate a local (in p) optimization approach to the design of lightpath routing, which is relatively easy compared with an optimization over all the values of p. We begin with the following definition:

Definition 4.1 *For given logical and physical topologies, a lightpath routing is said to be* uniformly optimal *if its reliability is greater than or equal to that of any other lightpath routing for every value of p.*

Therefore, a uniformly optimal lightpath routing yields the best reliability for any value of $p \in [0, 1]$. Based on the failure polynomial of a lightpath routing, one can immediately develop a sufficient condition for a uniformly optimal lightpath routing:

Theorem 4.1 *Given a lightpath routing R, let N_i^R be the number of cross-layer cuts with size i. Then R is a uniformly optimal lightpath routing if, for any other lightpath routing R', $N_i^R \leq N_i^{R'}$ for all $i \in \{0, \ldots, m\}$, where m is the number of physical links.*

Proof. The unreliability for the lightpath routings R and R' are given by:

$$\sum_{i=0}^{m} N_i^R p^i (1-p)^{m-i}, \quad \text{and}$$

$$\sum_{i=0}^{m} N_i^{R'} p^i (1-p)^{m-i}$$

respectively. It follows that:

$$\begin{aligned}
&\sum_{i=0}^{m} N_i^R p^i(1-p)^{m-i} - \sum_{i=0}^{m} N_i^{R'} p^i(1-p)^{m-i} \\
=&\sum_{i=0}^{m} (N_i^R - N_i^{R'}) p^i(1-p)^{m-i} \\
\leq& 0,
\end{aligned}$$

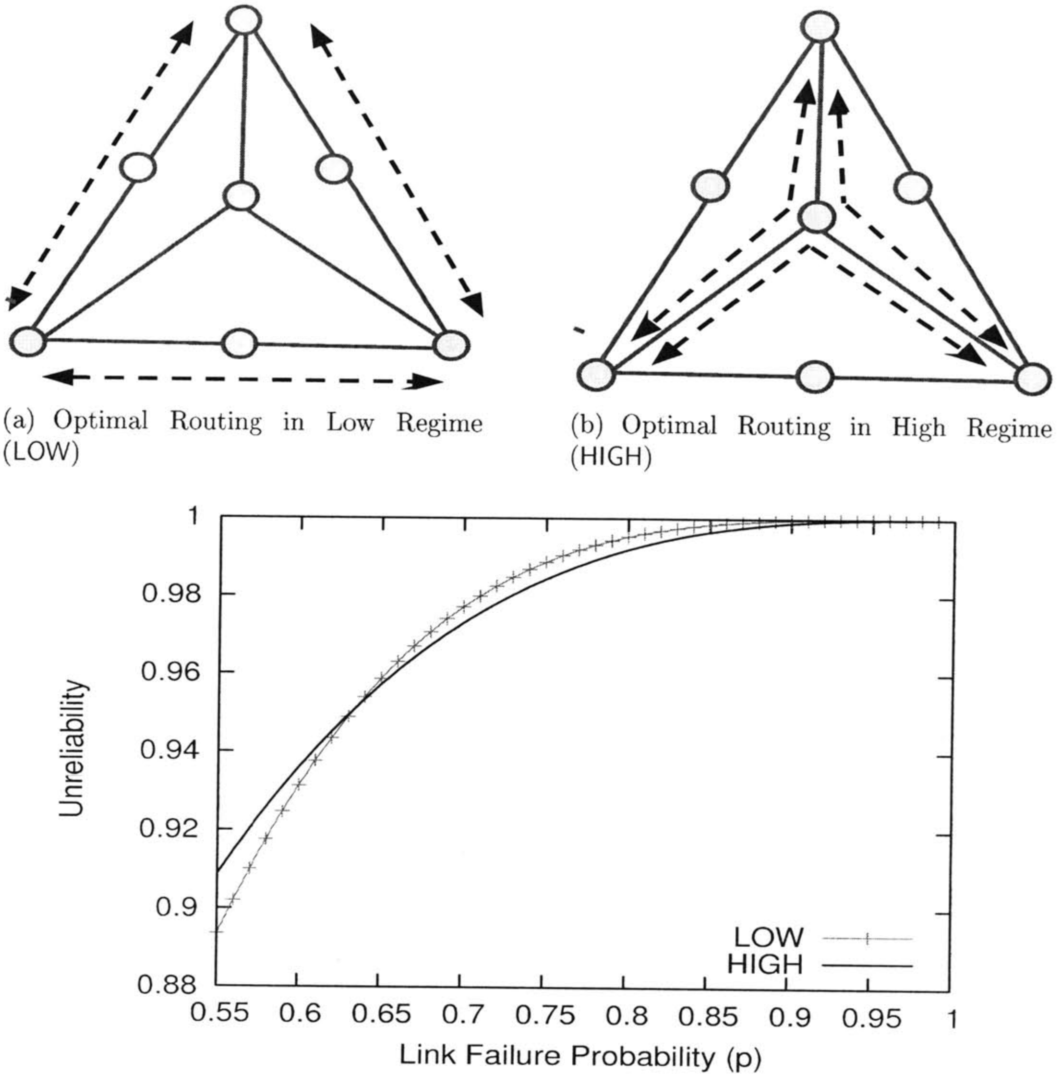
for any $p \in [0, 1]$, which implies that the reliability for R is always no less than any other lightpath routings. □

The existence of a uniformly optimal lightpath routing depends on the logical and physical topologies. For example, the lightpath routing shown in Figure 3-1(e) is uniformly optimal for the topologies in Figure 3-1. In contrast, Figure 4-1 shows two different lightpath routings that are optimal when p is sufficiently small and sufficiently large, respectively. In this case, there is no single lightpath routing which yields the highest reliability regardless of the link failure probability. Also note that in Figure 4-1(a), all the logical links are routed with physically disjoint paths that also happen to be physically shortest paths. Therefore, a lightpath routing that uses both physically shortest *and* disjoint paths does not guarantee uniform optimality in general. However, we conjecture that the following special class of *single-hop lightpath routing* is uniformly optimal:

Conjecture 1 *Given a physical topology $G_P = (V_P, E_P)$, and logical topology $G_L = (V_L, E_L)$ where $E_L \subseteq E_P$, the single-hop lightpath routing, where each logical link (s, t) takes on the physical fiber (s, t) as its physical route, is uniformly optimal.*

Since uniformly optimal lightpath routings are not always attainable, this motivates us to focus on non-uniformly (or locally) optimal routings, where the probability regime of optimality is restricted to a subrange within $[0, 1]$. A locally optimal lightpath routing is defined as follows:

Definition 4.2 *For given logical and physical topologies, a lightpath routing is said to be* locally optimal *if there exists $0 \leq a < b \leq 1$, such that its reliability is greater*



(a) Optimal Routing in Low Regime (LOW)

(b) Optimal Routing in High Regime (HIGH)

(c) Unreliability of the two Lightpath Routings

Figure 4-1: Example showing that a uniformly optimal routing does not always exist. Physical topology is in solid line, logical topology is the triangle formed by the 3 corner nodes and 3 edges, and lightpath routing is in dashed line.

than or equal to that of any other lightpath routing for every value of $p \in [a, b]$. In addiiton, the interval $[a, b]$ is called the optimality regime for the lightpath routing.

Note that a uniformly optimal lightpath routing is also locally optimal with optimality regime $[0, 1]$. Theorem 4.2 below is a crucial result to this study; namely, it reveals a connection between local optimality and uniform optimality.

Theorem 4.2 *Consider a pair of logical and physical topologies (G_L, G_P) for which there exists a uniformly optimal lightpath routing. Then, any locally optimal lightpath* *routing for (G_L, G_P) is also uniformly optimal.*

Proof. Denote by $F^*(p)$ the failure polynomial of a uniformly optimal lightpath routing. By definition, $F^*(p)$ is no greater than any other failure polynomial for $p \in [0, 1]$. Consider a locally optimal lightpath routing L, and let $F^L(p)$ be its failure polynomial. Let $[p_1, p_2]$ be the interval over which the routing L is optimal.

The polynomial equation $F^L(p) - F^*(p) = 0$ has degree at most m and thus has at most m roots unless the polynomial $F^L(p) - F^*(p)$ is trivially zero. However, by the definitions of local optimality and uniform optimality, the equation has an infinite number of solutions over the interval $[p_1, p_2]$. Consequently, $F^L(p)$ is identical to $F^*(p)$, which implies that lightpath routing L is also uniformly optimal. □

Motivated by this result, we study locally optimal lightpath routings. In particular, we develop the conditions for a lightpath routing to be optimal for both the low failure probability regime (small p) and high failure probability regime (large p).

4.4.2 Low Failure Probability Regime

It is easy to see that in the failure polynomial, the terms corresponding to small cross-layer cuts dominate when p is small. Hence, for reliability maximization in the low failure probability regime, it is desirable to minimize the number of small cross-layer cuts. We use this intuition to derive the properties of optimal routings for small p. We begin with the following definition:

Definition 4.3 *Consider two lightpath routings* 1 *and* 2. *Routing* 1 *is said to be* more reliable *than routing* 2 in the low failure probability regime *if there exists a positive number p_0 such that the reliability of routing* 1 *is higher than that of routing* 2 *for $0 < p < p_0$. A lightpath routing is said to be* locally optimal in the low failure probability regime *if it is more (or equally) reliable than any other routing in the low failure probability regime.*

Let d_j be the size of the MCLC under routing $j (= 1, 2)$. Let N_i and M_i be the numbers of cross-layer cuts of size i under routings 1 and 2 respectively. We call the vector $N = [N_i, \forall i]$ the *cut vector*. The following is an example of cut vectors N and M with $d_1 = 4$ and $d_2 = 3$:

i	0	1	2	3	4	5	$\cdots$	m
N_i	0	0	0	0	20	26	$\cdots$	1
M_i	0	0	0	9	19	30	$\cdots$	1.

Using cut vectors of lightpath routings, we define *lexicographical ordering* as follows:

Definition 4.4 *Routing* 1 *is lexicographically smaller than routing* 2 *if* $N_c < M_c$ *where* c *is the smallest* i *at which* N_i *and* M_i *differ.*

In the above example, we have $c = 3$ and $N_c < M_c$, hence routing 1 is lexicographically smaller. Therefore, if a lightpath routing is lexicographically smaller than another, it has fewer small cross-layer cuts and thus yields better reliability for small p.

Theorem 4.3 *Given two lightpath routings 1 and 2 with cut vectors* $[N_i | i = 0, \dots, m]$ *and* $[M_i | i = 0, \dots, m]$ *respectively, where* m *is the number of physical links, if routing* 1 *is lexicographically smaller than routing* 2*, then routing* 1 *is more reliable than routing* 2 *in the low failure probability regime. In particular, let* $c = \min_{0 \le i \le m} \{i : M_i \neq N_i\}$ *be the index where the elements in the cut vectors first differ. There exists* $p_0 \geq \frac{(c+1)(M_c - N_c)}{2m\binom{m}{c}}$ *such that lightpath routing 1 is more reliable than routing 2 for* $p < p_0$.

Proof. This is implied by Theorem 4.11, which will be proved in Section 4.5. □

Clearly, Theorem 4.3 leads to a local optimality condition; that is, if a lightpath routing minimizes the cut vector lexicographically, then it is locally optimal in the low failure probability regime. An interesting case is when routing 1 has larger MCLC than routing 2 (as in the above example). In this case, routing 1 is lexicographically smaller than routing 2 and implies Theorem 3.11, which we restate here as a corollary:

Corollary 4.4 *If* $d_1 > d_2$*, then routing* 1 *is more reliable than routing* 2 *in the low failure probability regime.*

Consequently, a lightpath routing with the maximum size MCLC yields the best reliability for small p. Similarly, routing 1 is also lexicographically smaller than routing 2 when they have the same size of MCLC but routing 1 has fewer MCLCs. This leads to the following result:

Corollary 4.5 *If $d_1 = d_2$ and $N_{d_1} < M_{d_2}$, then routing 1 is more reliable than routing 2 in the low probability regime.*

The expression for p_0 given in Theorem 4.3 also provides some insight into how the difference of the cut vectors affects the guaranteed regime. For example, if c is small and $M_c - N_c$ is large, the guaranteed regime is larger. In other words, if one lightpath routing has fewer small cross-layer cuts than the other, it will achieve higher reliability for a larger range of p in the low probability regime.

Therefore, for reliability maximization in the low failure probability regime, it is desirable to maximize the size of the MCLC while minimizing the number of such MCLCs. This condition will be used to develop lightpath routing algorithms in Chapter 5.

Finally, Theorem 4.3 also implies that all lightpath routings that are locally optimal in the low failure probability regime have the same failure polynomial. In other words, from the reliability standpoint, all locally optimal lightpath routings in the low failure probability regime are equivalent.

Corollary 4.6 *Let A and B be two different locally optimal lightpath routings in the low failure probability regime. Then the reliability of the two lightpath routings are identical, for all link failure probability p.*

Proof. We show that the failure polynomials of the two lightpath routings are identical. Suppose the failure polynomials are different. Then one of the lightpath routings is lexicographically smaller than the other. Therefore, one of them cannot be locally optimal in the low failure probability regime. □

4.4.3 High Failure Probability Regime

We have seen that when p is small, it is important to minimize the number of small cuts. Analogously, for large p, large cuts are dominant, and hence, minimizing the number of large cuts would result in maximum reliability. In other words, the cut vector should be minimized for large cuts for better reliability in the high failure probability regime. Similar to the case of low probability regime, we define the following:

Definition 4.5 *Consider two lightpath routings* 1 *and* 2. *Routing* 1 *is said to be* more reliable *than routing* 2 in the high failure probability regime *if there exists a number* $p_0 < 1$ *such that the reliability of routing* 1 *is higher than that of routing* 2 *for* $p_0 < p$.

An important parameter in this case is the Max Cross Layer Non-Cut (MCLNC), because logical networks with large MCLNC may remain connected even if only a small number of physical links survive. For high failure probability regime, the *colexicographical ordering* of the lightpath routings can be used to compare reliability performance. A cut vector $[N_i|i = 0, \dots, m]$ is colexicographically smaller than another cut vector $[M_i|i = 0, \dots, m]$ if and only if the vector $[N_{m-i}|i = 0, \dots, m]$ is lexicographically smaller than $[M_{m-i}|i = 0, \dots, m]$. In other words, rather than based on the first element in the vectors that differ, the colexicographical ordering is based on the last element in the vectors that differ. Therefore, if a lightpath routing has a larger MCLNC, it is also colexicographically smaller. The following theorem is a similar result to Theorem 4.3.

Theorem 4.7 *Given two lightpath routings 1 and 2 with cut vectors* $[N_i|i = 0, \dots, m]$ *and* $[M_i|i = 0, \dots, m]$ *respectively, where* m *is the number of physical links, if routing* 1 *is colexicographically smaller than routing* 2, *then routing* 1 *is more reliable than routing* 2 *in the high failure probability regime. In particular, let* $c = \max_{0 \le i \le m} \{i : M_i \neq N_i\}$ *be the index where the elements in the cut vectors last differ. There exists* $p_0 \le 1 - \frac{(m-c+1)(M_c - N_c)}{2m\binom{m}{c}}$ *such that lightpath routing 1 is more reliable than routing 2 for* $p > p_0$.

Proof. This is implied by Theorem 4.13, which will be proved in Section 4.5. □

Let c_j be the size of MCLNC for routing $j (= 1, 2)$. We can develop the following corollaries similar to the low regime case:

Corollary 4.8 *If $c_1 > c_2$, then routing 1 is more reliable than routing 2 in the high failure probability regime.*

Corollary 4.9 *If $c_1 = c_2$ and $N_{c_1} < M_{c_2}$, then routing 1 is more reliable than routing 2 in the high failure probability regime.*

Corollary 4.10 *Let A and B be two different locally optimal lightpath routings in the high failure probability regime. Then the reliability of the two lightpath routings are identical, for all link failure probability p.*

Therefore, for reliability maximization in the high failure probability regime, it is desirable to find a lightpath routing that maximizes the size of MCLNC (or equivalently, minimizes the size of MCLST) and minimizes the number of MCLNCs (or maximizes the number of MCLST). This observation is similar to the single-layer setting where maximizing the number of spanning trees maximizes the reliability for large p [16]. The major difference in the multi-layer case is that, since spanning trees may have different sizes, minimizing the *size* of the Min Cross-Layer Spanning Tree becomes the primary objective. As shown in Section 2.2.3, computing the size of the MCLST is NP-hard. Therefore, designing a lightpath routing that minimizes the MCLST is likely to be a difficult problem. In Appendix 4.8.1, we present an ILP that formulates the survivable lightpath routing problem with an objective to minimize the MCLST.

4.5 Extension of Probability Regimes

In the previous sections we have shown that a lightpath routing with a cut vector that is lexicographically (or colexicographically) smaller will have a higher reliability when link failure probability is sufficiently small (or high). However, the guaranteed regimes established in Theorems 4.3 and 4.7 are usually rather conservative, since the expressions only consider the the first element in the two cut vectors that are different. For instance, the expression fails to capture the uniform optimality for a lightpath routing that satisfies the condition in Theorem 4.1. In this section, we will develop a more general expression for the regime bounds that includes other elements in the cut vectors.

Consider two lightpath routings 1 and 2. Let $F_j(p)$ be the failure polynomial of routing j $(= 1, 2)$, and N_i's and M_i's be the coefficients in $F_1(p)$ and $F_2(p)$ respectively. Define the following two vectors of partial sums:

$$\overrightarrow{\mathbf{N}} = \left[\sum_{i=0}^{k} N_i | k = 0, ..., m\right] \text{ and } \overleftarrow{\mathbf{N}} = \left[\sum_{i=m-k}^{m} N_i | k = 0, ..., m\right].$$

The vectors $\overrightarrow{\mathbf{M}}$ and $\overleftarrow{\mathbf{M}}$ are defined similarly. Note that the i-th element $\overrightarrow{N}_i$ of vector $\overrightarrow{\mathbf{N}}$ is the total number of cross-layer cuts of size at most i. Likewise, $\overleftarrow{N}_i$ is the total number of cross-layer cuts of size at least i. We will use these vectors to develop the conditions that incrementally include larger cuts and thus extend the probability regime where one lightpath routing is more reliable than any other. We first extend the defintion of *lexicographical ordering* as follows:

Definition 4.6 *Lightpath routing* 1 *is said to be k-lexicographically smaller than lightpath routing* 2 *if*

$$k = \max\left\{j : \overrightarrow{N}_i \leq \overrightarrow{M}_i, \quad \forall i < d + j\right\} \text{ and } k \geq 1,$$

where d is the position of first element where the two cut vectors differ.

Therefore, a lightpath routing is lexicographically smaller (in the original sense) if and only if it is k-lexicographically smaller for some $k \geq 1$. The k-lexicographical ordering thus compares two lightpath routings based on structures beyond the smallest cuts, making it possible to establish a larger optimality regime. Roughly speaking, the value of k reflects the degree of dominance of a lightpath routing in the low probability regime: a k-lexicographically smaller lightpath routing means that it has fewer "small" cuts, where the definition for "small" is broader if k is larger.

Similarly, for the high failure probability regime, the colexicographical ordering defined in Section 4.4.3 can be extended to compare cuts beyond only the largest cuts:

Definition 4.7 *Lightpath routing* 1 *is said to be k-colexicographically smaller than lightpath routing* 2 *if*

$$k = \max\left\{j : \overleftarrow{N}_i \leq \overleftarrow{M}_i, \quad \forall i > c - j\right\} \text{ and } k \geq 1,$$

where c is the position of last element where the two cut vectors differ.

In contrast to the k-lexicographical ordering, this colexicographical ordering starts from the largest cuts, and incrementally includes the smaller cuts.

It is obvious that when $p \leq 0.5$, the failure probability of a cross-layer cut is a non-increasing function of the cut size, because $p^i(1-p)^{m-i} \geq p^{i+1}(1-p)^{m-(i+1)}$ for $p \leq 0.5$. Suppose that routing 1 has smaller total number of cuts of size up to i than routing 2, i.e., $\overrightarrow{N}_i \leq \overrightarrow{M}_i$. To compare cross-layer cuts of size at most $i+1$, suppose further that the relative increment $N_{i+1} - M_{i+1}$ in the number of larger cuts does not exceed the surplus $\overrightarrow{M}_i - \overrightarrow{N}_i$ from smaller cuts, i.e., $\overrightarrow{N}_{i+1} \leq \overrightarrow{M}_{i+1}$. Then, with respect to cut size at most $i+1$, routing 1 will have smaller failure probability than routing 2, provided that the same was true for cut size up to i. This observation leads to the following theorem on the relationship between lexicographical ordering and probability regime.

Theorem 4.11 *Given two vectors* $\mathbf{N}=[N_i|i=0,\ldots,m]$ *and* $\mathbf{M}=[M_i|i=0,\ldots,m]$. *For any j, let* $\overrightarrow{\Delta}_j = \sum_{i=0}^{j}(M_i - N_i)$ *and* $\overrightarrow{\delta}_j = \max_{j+1 \leq i \leq m}\left\{\frac{N_i - M_i}{\binom{m}{i}}\right\}$. *If the vector* $\mathbf{N}$ *is k-lexicographically smaller than* $\mathbf{M}$, *then:*

$$\sum_{i=0}^{m} N_i p^i (1-p)^{m-i} \leq \sum_{i=0}^{m} M_i p^i (1-p)^{m-i},$$

for $p \leq p_0^l = \min\left\{0.5, \max_{d \leq j \leq d+k-1} B_j\right\}$*, where* $d = \min\{i : N_i < M_i\}$ *and:*

$$B_j = \begin{cases} 0.5, & \text{if } j = m \\ \frac{1}{\frac{m}{j+1} + \overrightarrow{\delta}_j \binom{m}{j+1} / \overrightarrow{\Delta}_j}, & \text{otherwise.} \end{cases}$$

Proof. See Appendix 4.8.2. □

Therefore, the probability regime in Theorem 4.11 is a non-decreasing function of k, which means that a lightpath routing with smaller number of cuts over a larger size range will be guaranteed to be more reliable over a larger regime. This is consistent with the conclusion in Section 4.4.2, that the lightpath routing design should minimize the lexicographical ordering of the cut vector.

Theorem 4.3 is a direct result from Theorem 4.11. For a lexicographically smaller lightpath routing, the term B_d in Theorem 4.11 is given by:

$$\begin{aligned} \frac{1}{\frac{m}{d+1} + \overrightarrow{\delta}_d \binom{m}{d+1} / \overrightarrow{\Delta}_d} &= \frac{1}{\frac{m}{d+1} + \overrightarrow{\delta}_d \binom{m}{d+1} / (M_d - N_d)} \\ &\geq \frac{(d+1)(M_d - N_d)}{m(M_d - N_d) + (d+1)\binom{m}{d+1}}, \quad \text{since } \overrightarrow{\delta}_d \leq 1 \\ &\geq \frac{(d+1)(M_d - N_d)}{m\binom{m}{d} + (m-d)\binom{m}{d}} \\ &\geq \frac{(d+1)(M_d - N_d)}{2m\binom{m}{d}}. \end{aligned}$$

An interesting special case is when $d + k - 1 = m$, that is, $\overrightarrow{M}_j \geq \overrightarrow{N}_j$ for all $j = 0, \ldots, m$. In that case, the term $B_{d+k-1} = B_m = 0.5$, implying that the optimality regime is $[0, 0.5]$. We summarize this as the following corollary:

Corollary 4.12 *If* $\overrightarrow{N}_j \leq \overrightarrow{M}_j$ *for all* $j = 0, \ldots, m$*, then lightpath routing* 1 *is at least as reliable as lightpath routing* 2 *for* $p \leq 0.5$*, i.e.,* $F_1(p) \leq F_2(p)$ *for* $p \leq 0.5$.

Note that the condition in Corollary 4.12 requires every partial sum in the vector **M** to be at least the corresponding partial sum in the vector **N**, which is a much stronger condition than the lexicographic comparison in Theorem 4.3. This stronger condition allows the better optimality regime to be established in Corollary 4.12.

For the high failure probability regime, the result is similar to Theorem 4.11:

Theorem 4.13 *Given two vectors* $\mathbf{N}=[N_i|i=0,\dots,m]$ *and* $\mathbf{M}=[M_i|i=0,\dots,m]$. *For any* j, *let* $\overleftarrow{\Delta}_j = \sum_{i=m-j}^{m} (M_i - N_i)$ *and* $\overleftarrow{\delta}_j = \max_{0\leq i\leq m-j-1}\left\{\frac{N_i-M_i}{\binom{m}{i}}\right\}$. *If* $\mathbf{N}$ *is* k-*colexicographically smaller than* $\mathbf{M}$, *then:*

$$\sum_{i=0}^{m} N_i p^i (1-p)^{m-i} \leq \sum_{i=0}^{m} M_i p^i (1-p)^{m-i},$$

for $p \geq p_0^h = 1 - \max\left\{0.5, \min_{c\leq j\leq c+k-1} C_j\right\}$, *where* $c = \min\{i : N_{m-i} < M_{m-i}\}$ *and:*

$$C_j = \begin{cases} 0.5, & \textit{if } j = m \\ 1 - \frac{1}{\frac{m}{j+1} + \overleftarrow{\delta}_j \binom{m}{j+1} / \overleftarrow{\Delta}_j}, & \textit{otherwise.} \end{cases}$$

Proof. The proof for Theorem 4.13 is based on Theorem 4.11 and the symmetry between the k-lexicographical and k-colexicographical orderings. See Appendix 4.8.3 for details. □

The following corollary is analogous to Corollary 4.12 for the high failure regime:

Corollary 4.14 *If* $\overleftarrow{N}_j \leq \overleftarrow{M}_j$ *for all* $j = 0,\dots,m$, *then routing* 1 *is at least as reliable as routing* 2 *for* $p \geq 0.5$, *i.e.,* $F_1(p) \leq F_2(p)$ *for* $p \geq 0.5$.

Finally, combining Corollaries 4.12 and 4.14, this gives us a condition for uniformly optimal lightpath routing:

Corollary 4.15 *If* $\overrightarrow{N}_j \leq \overrightarrow{M}_j$ *and* $\overleftarrow{N}_j \leq \overleftarrow{M}_j$ *for all* $j = 0,\dots,m$, *then lightpath routing 1 is uniformly optimal.*

Theorems 4.11 and 4.13 unify Theorems 4.1, 4.3, and 4.7 to provide a single optimality regime expression for lightpath routings that exhibit different degrees of dominance. Note that the conditions of (co)lexicographical ordering in Corollaries 4.12 and 4.14 are satisfied by the uniform optimality condition $N_i \leq M_i, \forall i$ discussed in Theorem 4.1. Therefore, this unified theorem allows for a broader class of uniformly optimal lightpath routings.

4.6 Empirical Studies

In this section, we conduct empirical studies to verify the results presented in the previous sections. In Section 4.6.1, we study two sets of lightpath routings, optimized for the low and high failure probability regimes respectively, and compare their various attributes, in order to illustrate the structural difference between optimal lightpath routings for different regimes. We will also compare their reliability performance over the link failure probability regime $[0, 1]$. In Section 4.6.2, we compare the optimality regimes among the two sets of lightpath routings, and evaluate the tightness of the bounds given by Theorems 4.11 and 4.13.

All simulations in this section are based on the augmented NSFNET (Figure 4-2) with 14 nodes and 29 links as the physical topology, and 350 random logical topologies with size ranging from 6 to 12 nodes and connectivity at least 4. For our study of lightpath routings, we use the ILP-based rerouting algorithm that we will present in Section 5.1 to generate a set of lightpath routings, called $\mathsf{LPR}_{\mathsf{Low}}$, that are optimized for the low regime. Similarly, we use the formulation MCLST in Appendix 4.8.1 to generate a set of lightpath routings, called $\mathsf{LPR}_{\mathsf{High}}$, optimized for the high failure regime.

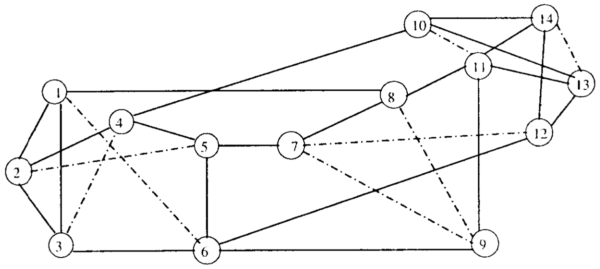


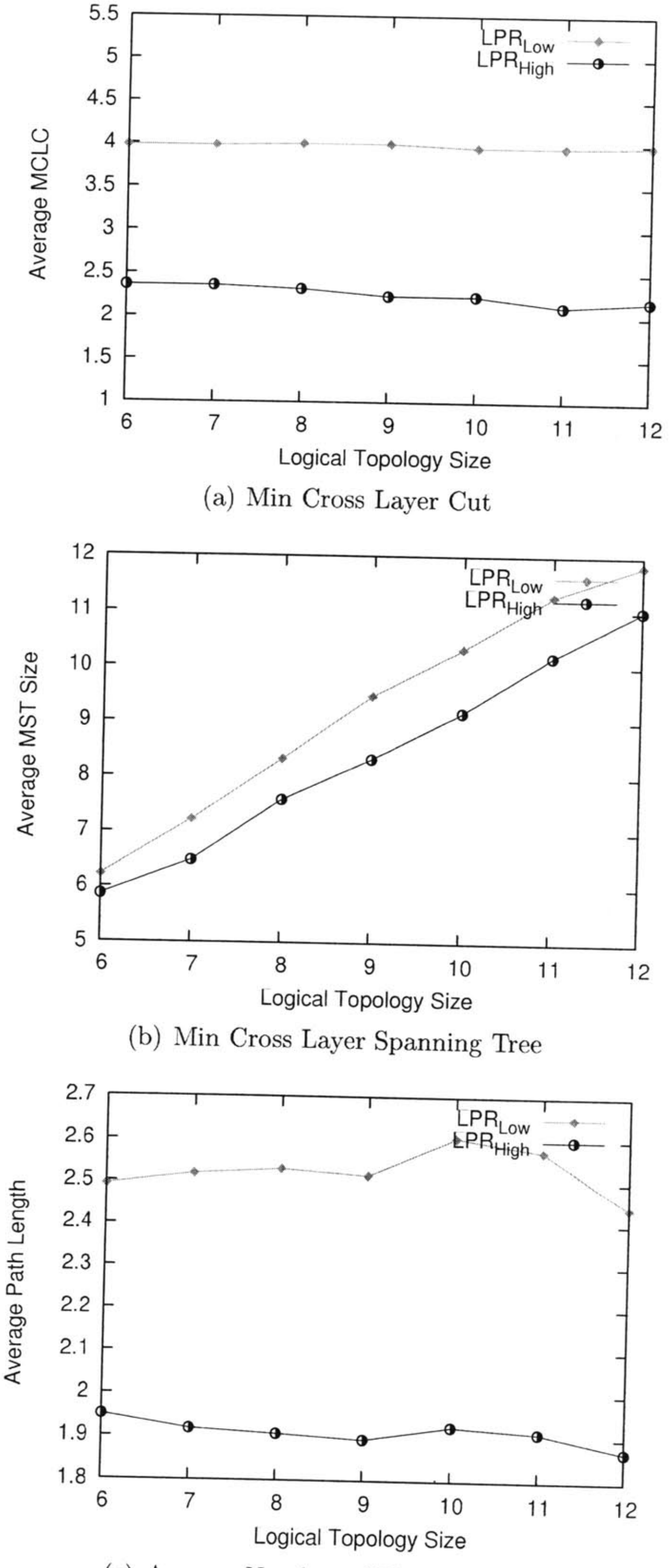
Figure 4-2: The augmented NSFNET.

4.6.1 Lightpath Routings Optimized for Different Probability Regimes

We first compare the structures of lightpath routings that are optimized for different failure probability regimes. Figures 4-3(a), 4-3(b) and 4-3(c) show the average values of MCLC, MCLST and the number of physical hops in the lightpaths for the two sets of lightpath routings $\mathsf{LPR}_{\mathsf{Low}}$ and $\mathsf{LPR}_{\mathsf{High}}$. For lightpath routings optimized for the high failure regime, the focus is to minimize the size of the minimum cross-layer spanning tree (MCLST), so it is not surprising that the size of the MCLST for $\mathsf{LPR}_{\mathsf{High}}$ is consistently smaller. As a side effect, minimizing the size of the MCLST often leads to shorter physical paths for the logical links, so the average number of physical hops for the logical links is consistently smaller for $\mathsf{LPR}_{\mathsf{High}}$ as well. On the other hand, the key to optimizing reliability for low failure regime is to maximize the MCLC, for which the lightpath routings in $\mathsf{LPR}_{\mathsf{Low}}$ are able to achieve better. Overall, there are noticeable differences in the structures between the two sets, suggesting that the two objectives can lead to vastly different lightpath routings.

In terms of reliability, this means that uniformly optimal lightpath routings may not always exist. In Figure 4-4, the survivability, both in terms of reliability and unreliability (i.e., 1 - reliability), of the pair over different link failure probabilities is shown. As expected, when the link failure probability is small, the lightpath routings in $\mathsf{LPR}_{\mathsf{Low}}$ achieve higher reliability. In particular, when the link failure probability approaches 0, there is an order of magnitude difference in terms of unreliability, meaning that maximizing the size of MCLC can have significant impact in the network reliability. As the link failure probability increases, it becomes more important to minimize the size of MCLST, so $\mathsf{LPR}_{\mathsf{High}}$ is able to achieve higher reliability in that regime.

Another interesting observation from the figure is that the difference in reliability is less prominent in the high failure probability regime. This is partly because the algorithm used to generate lightpath routings in $\mathsf{LPR}_{\mathsf{Low}}$, which tries to maximize the MCLC as well as minimize the number of MCLCs, is more sophisticated than



(a) Min Cross Layer Cut

(b) Min Cross Layer Spanning Tree

(c) Average Number of Physical Hops

Figure 4-3: Lightpath routings optimized for different probability regimes have different properties. LPR_{Low} are lightpath routings optimized for MCLC, and LPR_{High} are lightpath routings optimized for MCLST.

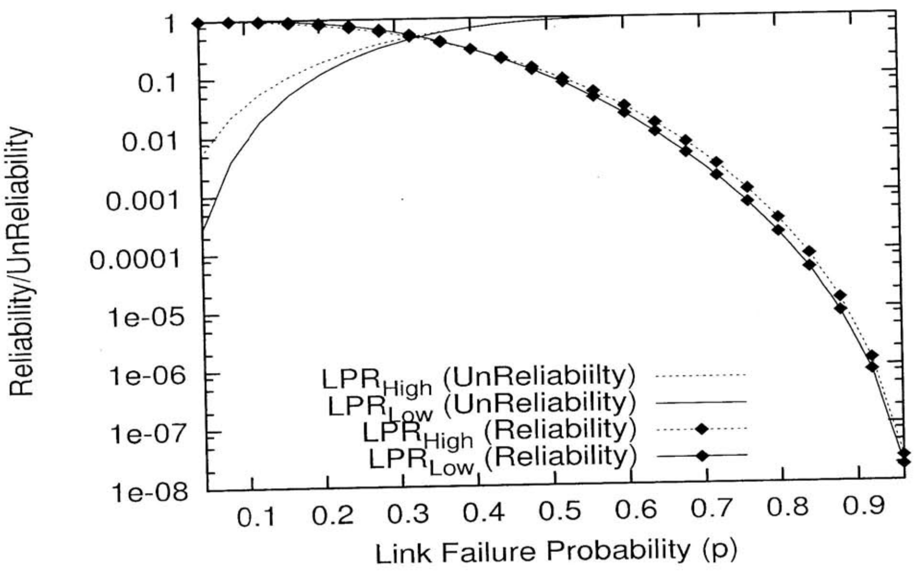


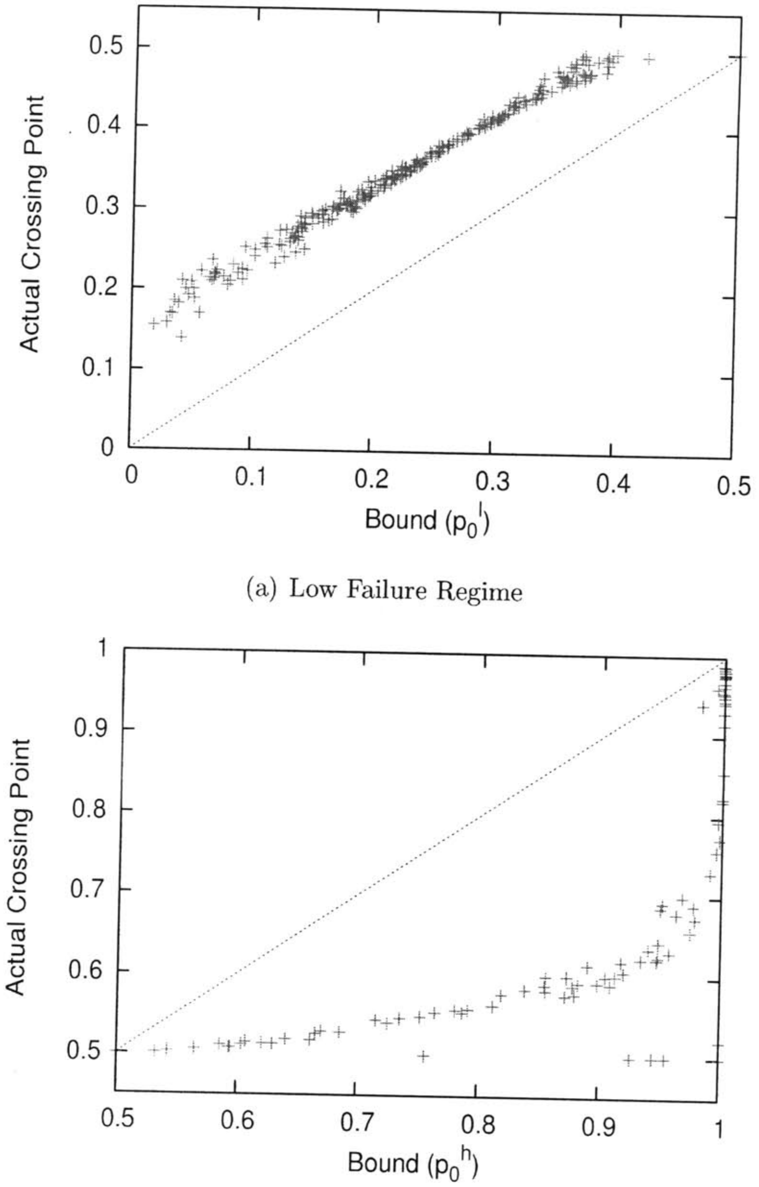
Figure 4-4: Reliability (or Unreliability) of lightpath routings optimized for different probability regimes.

the algorithm used to generate the lightpath routings in $\mathsf{LPR}_{\mathsf{High}}$. In addition, since the size of a MCLC is usually smaller than the size of a MCLST, the contribution of an MCLC to the unreliability in the low failure regime is generally greater than the contribution of a MCLST in the high failure regime. Therefore, the difference in reliability tends to be greater in the low failure probability regime.

In practical settings, the failure probability of individual physical links is typically very small. Therefore, our simulation result suggests that minimizing the lexicographic ordering of the lightpath routings can often lead to meaningful improvement in network survivability.

4.6.2 Bounds on Optimality Regimes

Next, we evaluate the bounds on optimality regimes, p_0^l and p_0^h, given by Theorems 4.11 and 4.13. For each pair of physical and logical topologies, we consider the corresponding lightpath routings in $\mathsf{LPR}_{\mathsf{Low}}$ and $\mathsf{LPR}_{\mathsf{High}}$. The values of p_0^l and p_0^h given by the theorems are compared with the actual crossing points of the failure polynomials, that is, the points where the (co)lexicographically smaller lightpath routings start to have lower reliability.



(a) Low Failure Regime

(b) High Failure Regime

Figure 4-5: Tightness of optimality regime bound. Each data point corresponds to the bound given by Theorems 4.11 and 4.13 vs the actual crossing point of the reliability polynomials of the two lightpath routings.

Each comparison corresponds to a data point in Figures 4-5(a) and 4-5(b), which plot the computed bounds against the actual crossing points for the two failure regimes. Since the bounds given by theorems are at most 0.5, for illustrative purpose the actual crossing points are also capped at 0.5.

In the low failure probability regime, there is a strong correlation between the value of p_0^l and the actual crossing point, suggesting that the bound provides a strong signal about the dominance of the lexicographically smaller lightpath routing in the low failure probability regime.

On the other hand, the correlation between the value of p_0^h and the actual crossing point is not as prominent in the high failure regime, meaning that the bounds are not as tight in this case. One possible explanation for this asymmetry is the difference in effectiveness between the algorithms used to generate the lightpath routings in $\mathsf{LPR}_{\mathsf{Low}}$ and $\mathsf{LPR}_{\mathsf{High}}$. As discussed before, the algorithm used to generate the lightpath routings in $\mathsf{LPR}_{\mathsf{Low}}$ is more sophisticated, and is able to generate solutions that are closer to the optimal. As a result, the lightpath routings in $\mathsf{LPR}_{\mathsf{Low}}$ generally exhibit a stronger dominance in the low failure probability regime, which results in tighter bounds given by Theorem 4.11. On the other hand, the lightpath routings in $\mathsf{LPR}_{\mathsf{High}}$ are less dominant in the high failure regime, which results in weaker bounds given by Theorem 4.13. This is confirmed by Figure 4-6, which shows the distribution of k in the k-(co)lexicographical ordering comparisons. Excluding the instances with total dominance, about 25% of the lightpath routings in $\mathsf{LPR}_{\mathsf{High}}$ are only 1-colexicographically smaller than their counterparts. In contrast, all the lightpath routings in $\mathsf{LPR}_{\mathsf{Low}}$ are at least 4-lexicographically smaller than their counterparts, so the bounds are tighter in general.

4.7 Conclusion

In this chapter, we study the relationship between the link failure probability, the cross-layer reliability and the structure of a layered network. The key to this study is the polynomial expression for reliability which relates structural properties of the network graph and the lightpath routing to the reliability. Using this polynomial, we show that reliable routings depend on the link failure probability, and identify optimality conditions for reliability maximization in different failure probability regimes. In particular, we show that a lightpath routing with the maximum size of Min Cross Layer Cuts (MCLC) and the minimum number of MCLCs is most reliable in the low

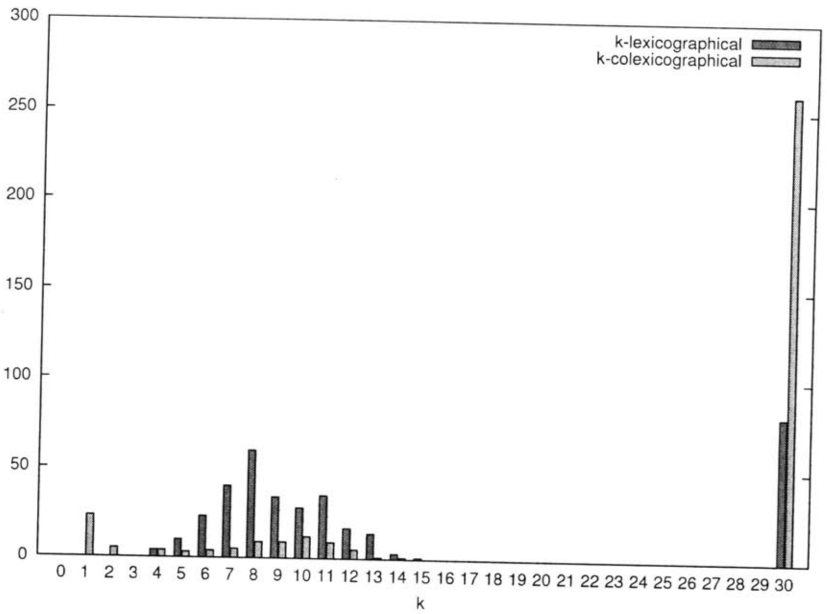


Figure 4-6: Histogram of k in k-(co)lexicographical ordering comparisons. Lightpath routings that dominate in every partial sum are put into the $k = 30$ bucket.

failure probability regime. On the other hand, in the high failure probability regime, a routing with the minimum size of Min Cross Layer Spanning Tree (MCLST) and the maximum number of MCLSTs maximizes reliability. This observation provides useful insights for designing reliable layered networks, which we will focus on in the next chapter.

4.8 Chapter Appendix

4.8.1 Lightpath Routing ILP to Minimize Minimum Cross Layer Spanning Tree (MCLST) Size

As discussed in Section 4.4.3, lightpath routings with smaller MCLST size will be more reliable in the high failure probability regime. In this section, we present an ILP for the lightpath routing formulation that minimizes the MCLST. This ILP is used in Section 4.6 to generate the set of lightpath routings, $\mathsf{LPR}_{\mathsf{High}}$, that are optimized for the high failure probability regime. We first define the following variables:

- $\{f_{ij}^{st} | (s,t) \in E_L, (i,j) \in E_P\}$: Flow variables representing the lightpath routing.

- $\{y_{ij} | (i,j) \in E_P\}$: 1 if fiber (i,j) survives, 0 otherwise.

- $\{z^{st} | (s,t) \in E_L\}$: 1 if lightpath (s,t) survives, 0 otherwise.

- $\{x^{st} | (s,t) \in E_L\}$: Flow variables on the logical topology.

$$\mathsf{MCLST}: \quad \text{Minimize} \sum_{(i,j)\in E_P} y_{ij}, \quad \text{subject to:}$$

$$\sum_{t\in V_L} x^{st} - \sum_{t\in V_L} x^{ts} = \begin{cases} |V_L| - 1, & \text{if } s = 0 \\ -1, & \text{if } s \in V_L - \{0\} \end{cases} \tag{4.2}$$

$$(V_L - 1) \cdot z^{st} \geq x^{st}, \quad \forall (s,t) \in E_L \tag{4.3}$$

$$y_{ij} \geq z^{st} + f_{ij}^{st} - 1 \quad \forall (s,t) \in E_L, \forall (i,j) \in E_P \tag{4.4}$$

$$\{(i,j) : f_{ij}^{st} = 1\} \text{ forms an } (s,t)\text{-path in } G_P,\ \forall (s,t) \in E_L$$

$$0 \leq y_{ij} \leq 1; \quad 0 \leq x^{st}; \quad z_{ij}, f_{ij}^{st} \in \{0,1\}$$

The variables x^{st} represent a flow on the logical topology where 1 unit of flow is sent from logical node 0 to every other logical node, as described by Constraint (4.2). Constraint (4.3) requires these flows to be carried only on the surviving logical links, which implies that the surviving links form a connected logical subgraph. Constraint (4.4) ensures the survival of physical links that are used by any surviving logical links. Since the objective function minimizes $\sum_{(i,j)\in E_P} y_{ij}$, the optimal solution will represent a minimum set of physical links whose survival will allow the logical link to be connected.

Therefore, the set of physical links (i,j) with $y_{ij} = 1$ forms a cross-layer spanning tree. As a result, the optimal solution to the above ILP yields a lightpath routing that minimizes the size of the MCLST.

4.8.2 Proof of Theorem 4.11

Theorem 4.11: Given two vectors $\mathbf{N}$=$[N_i | i = 0, \ldots, m]$ and $\mathbf{M}$=$[M_i | i = 0, \ldots, m]$. For any j, let $\overrightarrow{\Delta}_j = \sum_{i=0}^{j}(M_i - N_i)$ and $\overrightarrow{\delta}_j = \max_{j+1\leq i\leq m} \left\{ \frac{N_i - M_i}{\binom{m}{i}} \right\}$. If the vector $\mathbf{N}$ is

k-lexicographically smaller than $\mathbf{M}$, then:

$$\sum_{i=0}^{m} N_i p^i (1-p)^{m-i} \leq \sum_{i=0}^{m} M_i p^i (1-p)^{m-i},$$

for $p \leq p_0^l = \min\left\{0.5, \max_{d \leq j \leq d+k-1} B_j\right\}$, where $d = \min\{d : N_d < M_d\}$ and:

$$B_j = \begin{cases} 0.5, & \text{if } j = m \\ \frac{1}{\frac{m}{j+1} + \overrightarrow{\delta}_j \binom{m}{j+1} / \overrightarrow{\Delta}_j}, & \text{otherwise.} \end{cases}$$

Proof. We first prove the following lemma.

Lemma 4.16 *If vector* $\mathbf{N}$ *is* k*-lexicographically smaller than vector* $\mathbf{M}$*, then for all* $j \leq d+k-1$*, where* $d = \min\{d : N_d < M_d\}$*:*

$$\sum_{i=0}^{j} (M_i - N_i) p^i (1-p)^{m-i} \geq \overrightarrow{\Delta}_j p^j (1-p)^{m-j}, \quad \textit{for } 0 \leq p \leq 0.5. \tag{4.5}$$

Proof. We prove, by induction on j, that (4.5) holds for all $j \leq d+k-1$. First, if $j = 0$,

$$\begin{aligned} \sum_{i=0}^{j} (M_i - N_i) p^i (1-p)^{m-i} &= (M_0 - N_0)(1-p)^m \\ &= \overrightarrow{\Delta}_0 (1-p)^{m-j}. \end{aligned}$$

Therefore, (4.5) holds for $j = 0$. Now suppose (4.5) holds for all $i \leq j$ for some $j < d+k-1$. Then, we have:

$$\begin{aligned}
&\sum_{i=0}^{j+1}(M_i - N_i)p^i(1-p)^{m-i} \\
=&\sum_{i=0}^{j}(M_i - N_i)p^i(1-p)^{m-i} + (M_{j+1} - N_{j+1})p^{j+1}(1-p)^{m-(j+1)} \\
\geq&\overrightarrow{\Delta}_j p^j(1-p)^{m-j} + (M_{j+1} - N_{j+1})p^{j+1}(1-p)^{m-(j+1)}, \quad \text{by induction hypothesis} \\
\geq&\overrightarrow{\Delta}_j p^{j+1}(1-p)^{m-(j+1)} + (M_{j+1} - N_{j+1})p^{j+1}(1-p)^{m-(j+1)}, \quad \text{since } \tfrac{p}{1-p} \leq 1 \\
=&\overrightarrow{\Delta}_{j+1}p^{j+1}(1-p)^{m-(j+1)}.
\end{aligned}$$

Therefore, by induction, (4.5) is true for all $j \leq k$. □

Lemma 4.17 *Given a fixed k, if $\overrightarrow{\Delta}_i \geq 0$ for all $i \leq d+k-1$, then for any $d \leq j \leq d+k-1$:*

$$F_1(p) \leq F_2(p),$$

for $0 \leq p \leq \min\{0.5, B_j\}$, where:

$$B_j = \begin{cases} 0.5, & \text{if } j = m \\ \frac{1}{\frac{m}{j+1} + \overrightarrow{\delta}_j \binom{m}{j+1} / \overrightarrow{\Delta}_j}, & \text{otherwise.} \end{cases}$$

Proof. First, note that by definition of $\overrightarrow{\delta}_j$, for any $i \geq j$:

$$\overrightarrow{\delta}_j \binom{m}{i} \geq N_i - M_i. \tag{4.6}$$

If $k = m - d + 1$, then Lemma 4.16 implies that, for $p \leq 0.5$:

$$\begin{aligned}
\sum_{i=0}^{m}(M_i - N_i)p^i(1-p)^{m-i} &\geq \overrightarrow{\Delta}_m p^m \\
&\geq 0.
\end{aligned}$$

Therefore, the lemma is true for $k = m - d + 1$. Now suppose $k < m - d + 1$. If

$\overrightarrow{\delta}_j \leq 0$ for some $j \leq k$, this implies for any $d + k \leq l \leq m$:

$$\begin{aligned}
\overrightarrow{\Delta}_l &= \overrightarrow{\Delta}_{d+k-1} + \sum_{i=d+k}^{l} (M_i - N_i) \\
&\geq \overrightarrow{\Delta}_{d+k-1} - \sum_{i=d+k}^{l} \overrightarrow{\delta}_j \binom{m}{i}, \quad \text{by Equation (4.6)} \\
&\geq 0.
\end{aligned}$$

This last inequality is due to the fact that $\overrightarrow{\delta}_j \leq 0$, and that $\overrightarrow{\Delta}_{d+k-1} \geq 0$, since $\mathbf{N}$ is k-lexicographically smaller than $\mathbf{M}$. Therefore, in this case, the vector $\mathbf{N}$ is also $(m-d+1)$-lexicographically smaller than $\mathbf{M}$, and the lemma is true as proved above. Therefore, in the rest of the proof, we assume that $\overrightarrow{\delta}_j > 0$.

Since $p \leq 0.5$ and $\overrightarrow{\Delta}_i \geq 0$ for all $i \leq d + k - 1$, by Lemma 4.16 we have, for all $j \leq d + k - 1$:

$$\sum_{i=0}^{j} (M_i - N_i) p^i (1-p)^{m-i} \geq \overrightarrow{\Delta}_j p^j (1-p)^{m-j}. \tag{4.7}$$

Next, we will use the following result to bound the tail probability of the Binomial distribution:

Lemma 4.18 *For $r > mp$,*

$$\sum_{i=r}^{m} \binom{m}{i} p^i (1-p)^{m-i} \leq \binom{m}{r} p^r (1-p)^{m-r} \cdot \frac{r(1-p)}{r - mp}.$$

Proof. See [40]. □

Therefore, since $p \leq \frac{1}{\frac{m}{j+1} + \overrightarrow{\delta}_j \binom{m}{j+1} / \overrightarrow{\Delta}_j} < \frac{j+1}{m}$, by Lemma 4.18, we have:

$$\begin{aligned}
\sum_{i=j+1}^{m} \binom{m}{i} p^i (1-p)^{m-i} &\leq \binom{m}{j+1} p^{j+1} (1-p)^{m-(j+1)} \cdot \frac{(j+1)(1-p)}{j+1-mp} \\
&= \binom{m}{j+1} p^j (1-p)^{m-j} \cdot \frac{(j+1)p}{j+1-mp}.
\end{aligned} \tag{4.8}$$

In addition, since $p \le \frac{1}{\frac{m}{j+1}+\overrightarrow{\delta}_j\binom{m}{j+1}/\overrightarrow{\Delta}_j}$, we have:

$$\begin{aligned}
\frac{(j+1)p}{j+1-mp} &= \frac{1}{\frac{1}{p}-\frac{m}{j+1}} \\
&\le \frac{1}{\frac{\overrightarrow{\delta}_j\binom{m}{j+1}}{\overrightarrow{\Delta}_j}+\frac{m}{j+1}-\frac{m}{j+1}} \\
&= \frac{\overrightarrow{\Delta}_j}{\overrightarrow{\delta}_j\binom{m}{j+1}}.
\end{aligned} \tag{4.9}$$

It follows that:

$$\begin{aligned}
&\sum_{i=0}^{m}(M_i-N_i)p^i(1-p)^{m-i} \\
=&\sum_{i=0}^{j}(M_i-N_i)p^i(1-p)^{m-i}+\sum_{i=j+1}^{m}(M_i-N_i)p^i(1-p)^{m-i} \\
\ge&\sum_{i=0}^{j}(M_i-N_i)p^i(1-p)^{m-i}-\sum_{i=j+1}^{m}\overrightarrow{\delta}_j\binom{m}{i}p^i(1-p)^{m-i} \quad \text{by Equation (4.6)} \\
\ge&\overrightarrow{\Delta}_j p^j(1-p)^{m-j} \\
&\quad -\overrightarrow{\delta}_j\binom{m}{j+1}p^j(1-p)^{m-j}\cdot\frac{(j+1)p}{j+1-mp}, \quad \text{by Equations (4.7) and (4.8)} \\
=&p^j(1-p)^{m-j}\overrightarrow{\delta}_j\left(\frac{\overrightarrow{\Delta}_j}{\overrightarrow{\delta}_j}-\binom{m}{j+1}\cdot\frac{(j+1)p}{j+1-mp}\right) \\
\ge&p^j(1-p)^{m-j}\overrightarrow{\delta}_j\left(\frac{\overrightarrow{\Delta}_j}{\overrightarrow{\delta}_j}-\binom{m}{j+1}\cdot\frac{\overrightarrow{\Delta}_j}{\overrightarrow{\delta}_j\binom{m}{j+1}}\right), \quad \text{by Equation (4.9)} \\
=&0.
\end{aligned}$$

□

As a result of Lemma 4.17, we can pick the $d \le j \le d+k-1$ such that B_j is maximized to obtain the largest upper bound for p, and Theorem 4.11 follows. □

4.8.3 Proof of Theorem 4.13

Theorem 4.13: Given two vectors $\mathbf{N}{=}[N_i|i=0,\ldots,m]$ and $\mathbf{M}{=}[M_i|i=0,\ldots,m]$. For any j, let $\overleftarrow{\Delta}_j = \sum_{i=m-j}^{m}(M_i - N_i)$ and $\overleftarrow{\delta}_j = \max_{0\le i\le m-j-1}\left\{\frac{N_i-M_i}{\binom{m}{i}}\right\}$. If $\mathbf{N}$ is k-colexicographically smaller than $\mathbf{M}$, then:

$$\sum_{i=0}^{m} N_i p^i(1-p)^{m-i} \le \sum_{i=0}^{m} M_i p^i(1-p)^{m-i},$$

for $p \ge p_0^h = 1 - \max\left\{0.5, \min_{c\le j\le c+k-1} C_j\right\}$, where $c = \min\{i : N_{m-i} < M_{m-i}\}$ and:

$$C_j = \begin{cases} 0.5, & \text{if } j = m \\ 1 - \frac{1}{\frac{m}{j+1} + \overleftarrow{\delta}_j\binom{m}{j+1}/\overleftarrow{\Delta}_j}, & \text{otherwise.} \end{cases}$$

Proof. Let $N_i' = N_{m-i}$ and $M_i' = M_{m-i}$, for $i = 0,\ldots,m$; and let $\overrightarrow{N_k'} = \sum_{i=0}^{k} N_i'$ and $\overrightarrow{M_k'} = \sum_{i=0}^{k} M_i'$. It follows that the vector

$$\overrightarrow{N'} := [\overrightarrow{N_i'}|i=0,\ldots,m]$$

is k-lexicographically smaller than the vector

$$\overrightarrow{M'} := [\overrightarrow{M_i'}|i=0,\ldots,m].$$

By Theorem 4.11,

$$\begin{aligned}\sum_{i=0}^{m}(M_i - N_i)p^i(1-p)^{m-i} &= \sum_{i=0}^{m}(M_i' - N_i')q^i(1-q)^{m-i}, \quad \text{where } q = 1-p \\ &\ge 0,\end{aligned}$$

for $q \leq \min\left\{0.5, \max_{d \leq j \leq d+k-1} B_j\right\}$, where:

$$B_j = \begin{cases} 0.5, & \text{if } j = m \\ \frac{1}{\frac{m}{j+1} + \overrightarrow{\delta}'_j \binom{m}{j+1} / \overrightarrow{\Delta}'_j}, & \text{otherwise;} \end{cases}$$

In the above expression, we have:

$$\overrightarrow{\Delta}'_j = \sum_{i=0}^{j} M'_i - N'_i = \overleftarrow{\Delta}_j, \quad \text{and}$$
$$\overrightarrow{\delta}'_j = \max_{j+1 \leq i \leq m} \left\{ \frac{N'_i - M'_i}{\binom{m}{i}} \right\} = \overleftarrow{\delta}_j.$$

Note that $B_j = 1 - C_j$ for $d \leq j \leq d + k - 1$. Therefore, lightpath routing 1 is at least as reliable as lightpath routing 2 for

$$\begin{aligned} p =& 1 - q \\ \geq& 1 - \min\left\{0.5, \max_{d \leq j \leq d+k-1} B_j\right\} \\ =& \max\left\{0.5, \min_{d \leq j \leq d+k-1} 1 - B_j\right\} \\ =& \max\left\{0.5, \min_{d \leq j \leq d+k-1} C_j\right\}. \end{aligned}$$

□

Chapter 5

Algorithms to Improve Reliability in Layered Networks

In the previous chapter, we have shown that when physical link failures are rare, the lightpath routing that minimizes the lexicographical ordering will maximize the cross-layer reliability. We have proposed a number of survivable lightpath routing heuristics in Chapter 2 where the objective is to maximize the MCLC. Since a lightpath routing with a larger MCLC value is lexicographically smaller, these algorithms can be considered as the first step towards maximizing the cross-layer reliability under the low failure probability regime. In this chapter, we continue in this direction to develop algorithms that not only maximize the MCLC, but also minimize the number of MCLCs.

All algorithms developed in this chapter follow a common iterative pattern, where "local" changes are incrementally applied to the given layered network to improve its cross-layer reliability. In each iteration, some preprocessing is performed to construct the set of MCLCs in the network, and a local change to the network is applied such that at least some of these MCLCs will be eliminated after the change. The process is repeated until no further improvement can be found, in which case the lightpath routing reaches a local optimum lexicographically.

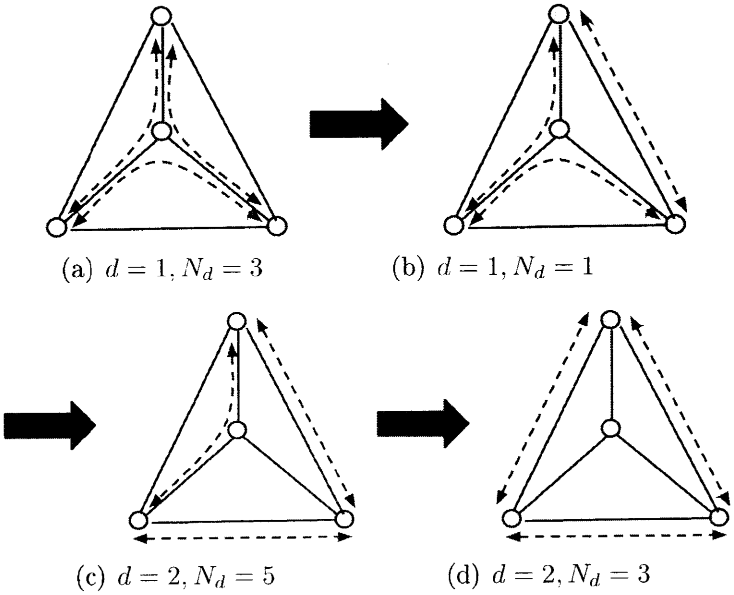
We will consider two different approaches under this framework. In Section 5.1, we will first study the *lightpath rerouting* method in which an iteration involves changing the physical route of an existing lightpath. By rerouting lightpaths in the network, one can possibly improve the reliability of a layered network without changing the physical and logical topologies. We will formulate the lightpath rerouting as an optimization problem, where the objective is to find best way to reroute a lightpath so that the reliability improvement is maximized. In Section 5.1.2, we will develop an ILP to find the optimal lightpath to reroute. In Section 5.1.3, we will propose an approximation algorithm that can compute a near-optimal solution in a much shorter time. Simulation results on these algorithms will be presented in Section 5.1.4.

Conceivably, one can further improve the reliability of the network by adding logical links to the network. Therefore, in Section 5.2, we will consider *logical topology augmentation* to improve the reliability of a layered network. By iteratively adding logical links to a network, one can eliminate some of the existing MCLCs of the network, thereby reducing the number of MCLCs, or potentially increasing the size of the MCLC. We will formulate the augmentation as an optimization problem, where the objective is to find the placement of the new logical link that will eliminate the largest number of MCLCs. Similar to the rerouting problem, an ILP and an approximation algorithm will be presented. In addition, in Section 5.2.5, we develop a lower bound on the minimum number of additional logical links required to increase the MCLC value of the layered network. We will use this lower bound to evaluate the effectiveness of our incremental augmentation algorithm.

Finally, to conclude this chapter, in Section 5.3 we will carry out a case study on a real-world IP-over-WDM network. We will apply different techniques developed throughout this thesis, including survivable lightpath routing, lightpath rerouting and logical topology augmentation to study the reliability gain achieved by these techniques in a real world setting.

5.1 Lightpath Rerouting

Given an existing lightpath routing of a layered network, the *lightpath rerouting* method involves changing the physical route of certain logical links in order to reduce the number of small cross-layer cuts in the network. Figure 5-1 shows a simple example of how rerouting can eliminate small cuts. In the figure, the solid lines depicts the physical topology and the dashed lines depicts the logical topology. Initially, the Min Cross Layer Cut size of the lightpath routing is 1 and there are three cross-layer cuts of this size. The logical links are then rerouted sequentially so that the network reliability is incrementally improved. At the end, the MCLC value of the lightpath routing is increased to 2.



(a) $d = 1, N_d = 3$

(b) $d = 1, N_d = 1$

(c) $d = 2, N_d = 5$

(d) $d = 2, N_d = 3$

Figure 5-1: Improving reliability via lightpath rerouting. The physical topology is in solid lines, and the lightpath routing of the logical topology is in dashed lines. The MCLC value and the number of MCLCs in the lightpath routings are denoted by d and N_d.

Generally speaking, the rerouting framework can be described as follows. Given any initial lightpath routing,

(1) Select a logical link, say (s, t), and reroute (s, t) to reduce the number of MCLCs.

(2) Repeat (1) until no further improvement is possible.

Therefore, each iteration will reduce the number of MCLCs, and possibly increase the size of the MCLC if every MCLC is converted into a non-cut. When the rerouting terminates, the final lightpath routing is *locally optimal*, in the sense that no further improvement is possible by rerouting a single lightpath.

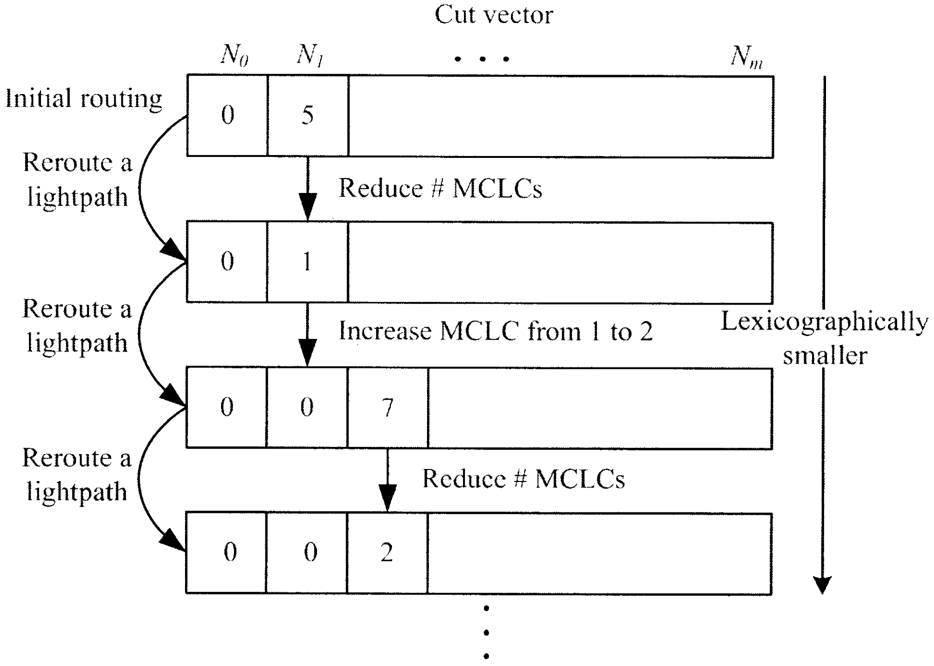


Figure 5-2: The lightpath rerouting framework.

In Chapter 2, we presented several formulations for routing the logical links jointly to maximize the MCLC. The lightpath rerouting framework provides an alternative approach for designing survivable lightpath routings. Instead of solving the formulations that jointly route the logical links, we can construct an initial lightpath routing using a fast algorithm such as the shortest path routing, and then iteratively apply rerouting until the lightpath routing reaches a local optimum. Since each iteration computes a physical route for only one logical link, this approach effectively breaks down the joint lightpath routing problem into multiple smaller steps, which helps improve the overall running time. As we will see in Section 5.1.4, this rerouting approach is very effective in obtaining lightpath routings with better reliability than the formulations in Chapter 2.

5.1.1 Effects of Rerouting a Lightpath

Suppose that an initial lightpath routing is given, and let d be the size of the MCLC under the initial routing. When the physical route of a logical link changes, some of the cross-layer cuts will be converted into non-cuts, and some non-cuts will be converted into cross-layer cuts. In the low failure probability regime, the reliability will be improved by the rerouting if the following is true:

1. The conversion of cross-layer cuts with size d to non-cuts outnumbers the conversion in the opposite direction.

2. The MCLC value does not decrease.

Therefore, we can formulate the lightpath rerouting as an optimization problem to maximize the reduction in the number of MCLCs, subject to the constraint that no non-cuts of size smaller than d is converted to cross-layer cuts. Here we will formulate such a reduction in the number of MCLCs by a lightpath rerouting, which will be used as the basis of the ILP formulation.

Given the physical topology $G_P = (V_P, E_P)$ and the logical topology $G_L = (V_L, E_L)$, we model a lightpath routing as a set of binary constants $\{f_{ij}^{st}\}$, where $f_{ij}^{st} = 1$ if and only if logical link (s,t) uses physical link (i,j) in the lightpath routing. For a given set of physical links S, we define the *logical residual graph* for S, denoted as G_L^S, to be $\left\{(s,t) \in E_L : \sum_{(i,j)\in S} f_{ij}^{st} = 0\right\}$. In other words, the residual graph consists of logical links that use none of the physical links in S. By definition, the set S is a cross-layer cut if and only if its logical residual graph is disconnected. Given a cross-layer cut S, it is called a *k-way cross-layer cut* if its logical residual graph has k connected components. In addition, given a cross-layer non-cut T for a lightpath routing, we call a logical link (s,t) *critical* to T if (s,t) is a cut edge of the residual graph G_L^T, that is, it is an edge in G_L^T whose removal will disconnect the residual graph.

The following theorems describe the conditions for a lightpath rerouting that results in conversions between cross-layer cuts and non-cuts.

Theorem 5.1 *Let S be a cross-layer cut for a lightpath routing. Rerouting logical link (s,t) from physical path P_1 to P_2 turns S into a non-cut if and only if the following conditions are true:*

1. S is a 2-way cross-layer cut.

2. *s and t are disconnected in the residual graph for S.*

3. *P_2 does not use any physical links in S.*

Proof. Let G_L^S and $G_L^{S'}$ be the residual logical graphs for S under the original and new lightpath routings respectively. First, suppose all the above conditions are true. Since S is a 2-way cross-layer cut under the original lightpath routing, the logical residual graph G_L^S consists of 2 connected components, each of which contains one of s and t. All logical links that are in G_L^S will remain in $G_L^{S'}$, because none of their physical routes have changed. In addition, since the new route P_2 does not use any physical links in S, the logical link (s,t) will be present in $G_L^{S'}$, making $G_L^{S'}$ connected. This implies S becomes a non-cut under the new lightpath routing.

Conversely, if S is a k-way cross-layer cut with $k > 2$, or s,t belong to the same connected component in G_L^S, rerouting (s,t) will not connect the logical residual graph, so S remains a cross-layer cut. In addition, if P_2 uses some physical link in S, (s,t) will not be present in the new residual graph $G_L^{S'}$, so $G_L^{S'} = G_L^S$, which also implies S remains a cross-layer cut. □

Theorem 5.2 *Let T be a cross-layer non-cut for a lightpath routing. Rerouting logical link (s,t) from physical path P_1 to P_2 turns T into a cross-layer cut if and only if the following conditions are true:*

1. *(s,t) is critical to T.*

2. *P_2 uses some physical link in T.*

Proof. Let G_L^T and $G_L^{T'}$ be the residual logical graphs for T under the original and new lightpath routings respectively. First, suppose both conditions are true. Since P_2 uses some physical link in T, the logical link will be removed from G_L^T under the new lightpath routing. Since (s,t) is critical to the non-cut T, its removal will disconnect the residual graph, which means that T will become a cross-layer cut.

Conversely, suppose any of the conditions are false. In this case, the logical residual graph $G_L^{T'}$ will remain connected after rerouting logical link (s,t). So T remains a non-cut. □

Therefore, the optimal rerouting should maximize the number of cross-layer cuts satisfying Theorem 5.1 and minimize the number of non-cuts satisfying Theorem 5.2. However, it is also important to ensure that none of the non-cuts with size smaller than d is converted to cross-layer cuts by the rerouting, since otherwise the MCLC value will decrease. The following theorem states that only non-cuts with size at least $d-1$ can be converted into a cross-layer cut by rerouting a single lightpath.

Theorem 5.3 *Let d be the Min Cross Layer Cut value of a lightpath routing and let $\mathcal{NC}$ be the set of cross-layer non-cuts that can be converted into cross-layer cuts by rerouting a single logical link. Then $|T| \geq d-1$ for all $T \in \mathcal{NC}$.*

Proof. Suppose $\mathcal{NC}$ contains a convertible non-cut T with size less than $d-1$. Since T is convertible by rerouting a single logical link, by Theorem 5.2, there exists a logical link (s,t) that is critical to T. Now let l be any physical link used by (s,t), then the set of physical links $T \cup \{l\}$ would disconnect the logical residual graph and is therefore a cross-layer cut. However, such a set contains at most $d-1$ physical links, contradicting that d is the Min Cross Layer Cut. □

Therefore, when rerouting a lightpath, we need to make sure that none of the non-cuts with size $d-1$ get converted into cuts in order to prevent the MCLC value from decreasing. Based on these observations, we next develop an ILP for the lightpath rerouting problem.

5.1.2 ILP for Lightpath Rerouting

Let (V_P, E_P) and (V_L, E_L) be the physical and logical topologies. For the given lightpath routing, let d be the MCLC value, and let $\mathcal{C}_d, \mathcal{NC}_d$ and $\mathcal{NC}_{d-1}$ be the sets of 2-way cross-layer cuts with size d, non-cuts with size d, and non-cuts with size $d-1$ respectively. The lightpath rerouting problem can be formulated as an ILP that finds the logical link, and its new physical route, that maximizes the net reduction in MCLCs.

The ILP can be considered as a path selection problem on an auxiliary graph $G'_P = (V'_P, E'_P)$, where $V'_P = V_P \cup \{u, v\}$, with u and v being the additional source and sink nodes in the auxiliary graph; and $E'_P = E_P \cup \{(u,x),(x,v) : x \in V_P\}$. Figure 5-3 illustrates the construction of the auxiliary graph.

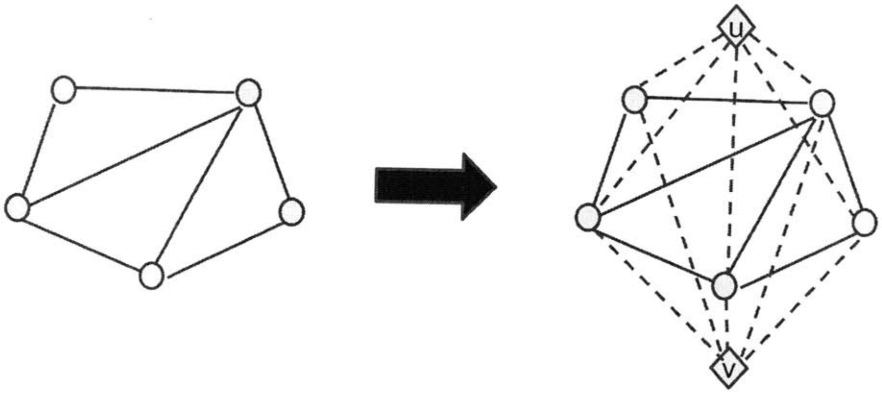


Figure 5-3: Construction of the auxiliary graph for the ILP. u and v are the additional source and sink nodes, and the dashed lines are the additional links in the auxiliary graph.

We first define the following variables and parameters:

1. Variables:

   - $\{g_{st} : (s,t) \in E_L\}$: 1 if logical link (s,t) is rerouted, and 0 otherwise.
   - $\{f_{ij} : (i,j) \in E'_P\}$: Flow variables describing a path in G' from node u to node v.
   - $\{y^c : c \in \mathcal{C}_d\}$: 1 if the cross-layer cut c is converted into a non-cut by the lightpath rerouting, and 0 otherwise.
   - $\{z^c : c \in \mathcal{NC}_d\}$: 1 if the non-cut c is converted into a cross-layer cut by the lightpath rerouting, and 0 otherwise.

2. Parameters:

   - $\{h^c_{st} : c \in \mathcal{C}_d, (s,t) \in E_L\}$: 1 if logical nodes s and t are disconnected by the 2-way cut c, and 0 otherwise.
   - $\{q^c_{st} : c \in \mathcal{NC}_d \cup \mathcal{NC}_{d-1}, (s,t) \in E_L\}$: 1 if logical link (s,t) is critical to the non-cut c, and 0 otherwise.
   - $\{l^c_{ij} : \forall c \in \mathcal{C}_d \cup \mathcal{NC}_d \cup \mathcal{NC}_{d-1}, (i,j) \in E_P\}$: 1 if physical link (i,j) is in set c, and 0 otherwise.

The lightpath rerouting can be formulated as follows:

$$\text{REROUTE}: \quad \text{Maximize} \sum_{c\in\mathcal{C}_d} y^c - \sum_{c\in\mathcal{NC}_d} z^c, \quad \text{subject to:}$$

$$g_{st} \le (f_{us} + f_{tv})/2, \quad \forall (s,t) \in E_L \tag{5.1}$$

$$\sum_{(s,t)\in E_L} g_{st} = 1 \tag{5.2}$$

$$l^c_{ij} f_{ij} + \sum_{(s,t)\in E_L} q^c_{st} g_{st} \le 1, \quad \forall c \in \mathcal{NC}_{d-1}, (i,j) \in E_P \tag{5.3}$$

$$l^c_{ij} f_{ij} + \sum_{(s,t)\in E_L} q^c_{st} g_{st} \le z^c + 1, \forall c \in \mathcal{NC}_d, (i,j) \in E_P \tag{5.4}$$

$$y^c \le \sum_{(s,t)\in E_L} h^c_{st} g_{st}, \quad \forall c \in \mathcal{C}_d \tag{5.5}$$

$$y^c \le 1 - l^c_{ij} f_{ij}, \quad \forall (i,j) \in E_P, \forall c \in \mathcal{C}_d \tag{5.6}$$

$$\{(i,j) : f_{ij} = 1\} \text{ forms an } (u,v)\text{-path in } G' \tag{5.7}$$

$$f_{ij}, g_{st} \in \{0,1\}, 0 \le y^c, z^c \le 1$$

The formulation can be interpreted as a path selection problem on the auxiliary graph G'_P. Constraint (5.7), which requires that the variables f_{ij} describe a path from u to v, can be expressed by the standard flow conservation constraints. As a result, in a feasible solution to the formulation, the variables f_{ij} represent a path $u \rightarrow s \rightsquigarrow t \rightarrow v$, which corresponds to the new physical route for the logical link (s,t) after the rerouting.

Constraint (5.1) ensures that g_{st} can be set to 1 only if f_{ij} represents the path $u \rightarrow s \rightsquigarrow t \rightarrow v$, and Constraint (5.2) makes sure that the chosen (s,t) is indeed a logical link in E_L. Therefore, exactly one logical link (s,t) can have $g_{st} = 1$, and a feasible solution to this ILP corresponds to a rerouting of the logical link.

In Constraint (5.3), the two terms correspond to the conditions in Theorem 5.3. The constraint makes sure that at most one of the conditions is satisfied, thereby disallowing the non-cuts of size $d-1$ to be converted into a cross-layer cut. Similarly, Constraint (5.4) makes sure $z^c = 1$ for any non-cut $c \in \mathcal{NC}_d$ that is converted into a cross-layer cut by the rerouting.

Finally, Constraints (5.5) and (5.6) describe conditions 2) and 3) of Theorem 5.1 respectively. Therefore, y^c can be 1 only if both conditions in the theorem are satisfied, which implies that cross-layer cut c is converted into a non-cut.

Since the objective is to maximize y^c and minimize z^c, in an optimal solution $y^c = 1$ if and only if cross-layer cut c is converted into a non-cut, and $z^c = 1$ if and only if non-cut c is converted into a cross-layer cut. As a result, the objective function reflects the net reduction in the number of MCLCs.

Note that the variables y^c and z^c will take on binary values in an optimal solution even if they are not constrained to be integral. This observation significantly reduces the number of binary variables in the formulation. There are $O(|E_P| + |E_L|)$ binary variables in the rerouting formulation, which is significantly less than the $O(|E_P||E_L|)$ binary variables in the Multi-Commodity Flow lightpath routing formulations in Chapter 2. As we will see in the simulation section, this translates to faster running time.

For larger networks, however, solving the rerouting ILP may still be infeasible in practice. One way to speed up the time to solve the ILP is to relax the binary variables f_{ij} in the formulation and use randomized rounding discussed in Section 2.4.3 to construct a (u, v)-path from the optimal solution of the relaxed formulation. In the following section, we describe a polynomial time d-approximation algorithm for the rerouting problem. This provides an alternative to apply rerouting in instances that are too large to solve the ILP optimally. We will evaluate the performance of all these approaches in Section 5.1.4.

5.1.3 An Approximation Algorithm for Lightpath Rerouting

We focus on the following question: Given the lightpath routing, and a logical link (s, t), what is the best way to reroute (s, t) assuming the routes for all other logical links are fixed? A solution to this problem will allow us to solve the lightpath rerouting problem, since we can run the algorithm once for each logical link, and return the best solution.

Similar to the previous section, let $\mathcal{C}_d, \mathcal{NC}_d$ and $\mathcal{NC}_{d-1}$ be the set of cross-layer cuts of size d, non-cuts of size d and non-cuts of size $d-1$ respectively. Now suppose Q is a new physical route for logical link (s,t). According to Theorem 5.2, a non-cut $T \in \mathcal{NC}_d \cup \mathcal{NC}_{d-1}$, will be converted into a cross-layer cut if and only if the following is true:

1. (s,t) is critical to T.

2. Q uses any physical links in T.

Let $\mathcal{NC}_d^{st}$ and $\mathcal{NC}_{d-1}^{st}$ be the subsets of $\mathcal{NC}_d$ and $\mathcal{NC}_{d-1}$ that satisfy condition (1). These two sets represent the non-cuts that can potentially be converted into a cut by rerouting (s,t). It immediately follows that any (s,t) path that uses a physical link in $\cup_{T \in \mathcal{NC}_{d-1}^{st}} T$ will create a cross-layer cut with size $d-1$, which should be forbidden for the new physical route. In addition, for any physical link (i,j), the set $\mathcal{L}_{ij}^{\mathcal{NC}} = \{T \in \mathcal{NC}_d^{st} : (i,j) \in T\}$ represents the non-cuts with size d that will be converted into cross-layer cuts if the new route Q for logical link (s,t) contains the physical link (i,j).

Similarly, for a cross-layer cut $S \in \mathcal{C}_d$, it will remain a cross-layer cut after the reroute if and only if any of the following is true, according to Theorem 5.1:

1. S is a k-way cut with $k > 2$.

2. s, t belong to the same connected component in the logical residual graph G_L^S.

3. Q uses any physical link that is contained in S.

Let $\mathcal{C}_d^{st} \subseteq \mathcal{C}_d$ be the set of cross-layer cuts that satisfy conditions (1) or (2). This represents the set that will continue to be cross-layer cuts regardless of the new physical route Q for (s,t). In addition, for each $(i,j) \in E_P$, the cross-layer cuts in the set $\mathcal{L}_{ij}^{\mathcal{C}} = \{S \in \mathcal{C}_d - \mathcal{C}_d^{st} : (i,j) \in S\}$ will also continue to be cross-layer cuts if the new route Q contains the physical link (i,j).

Now, for each physical link (i,j), let $\mathcal{L}_{ij} = \mathcal{L}_{ij}^{\mathcal{C}} \cup \mathcal{L}_{ij}^{\mathcal{NC}}$. If a physical link (i,j) is used by the logical link (s,t) in the new route Q, it will cause the set $\mathcal{L}_{ij} \cup \mathcal{C}_d^{st}$ to become cross-layer cuts. Since every set of physical links in $\mathcal{C}_d^{st}$ will be cross-layer cuts regardless of the physical route taken by (s,t), the lightpath rerouting problem for logical link (s,t) can be formulated as choosing the (s,t)-path Q in $G'_P = (V_P, E_P - \cup_{T\in\mathcal{NC}^{st}_{d-1}} T)$ that minimizes $|\cup_{(i,j)\in Q}\mathcal{L}_{ij}|$. Although this is an instance of the NP-Hard *Minimum Color Path* [124] problem, a simple d-approximation algorithm exists, as described below:

Algorithm 4 REROUTE_SP(s,t)

```
1: Construct a weighted graph on $G'_P = (V_P, E_P - \cup_{T\in\mathcal{NC}^{st}_{d-1}} T)$, where each edge
   $(i,j)$ is assigned with weight $w(i,j) = |\mathcal{L}_{ij}|$.
2: Run Dijkstra's algorithm to find the shortest $(s,t)$-path in the weighted graph.
```

We prove that REROUTE_SP is a d-approximation algorithm.

Theorem 5.4 *Let Q^* be the optimal physical route for (s,t) that results in the minimum number of MCLCs, and let Q^{SP} be the new route for (s,t) returned by* REROUTE_SP*. For any (s,t)-path Q, let $N_d(Q)$ be the number of cross-layer cuts with size d after rerouting (s,t) with Q, where d is the size of the MCLC. Then $N_d(Q^{\mathsf{SP}}) \leq d \cdot N_d(Q^*)$.*

Proof. Given any (s,t) path Q, define $\mathcal{L}(Q) = \cup_{(i,j)\in Q} L_{ij}$, it follows that $N_d(Q) = |\mathcal{L}(Q)| + |C^{st}_d| = |\mathcal{L}(Q)| + K$, where $K = |C^{st}_d|$ is a constant. In addition, let $w(Q)$ be the total weight sum of the path Q in the weighted graph constructed by REROUTE_SP(s,t).

Since each set of physical links $S \in \mathcal{L}(Q)$ has size d, we have $|\{(i,j) : S \in \mathcal{L}_{ij}\}| \leq d$, which implies:

$$
\begin{aligned}
w(Q) &= \sum_{(i,j)\in Q} |\mathcal{L}_{ij}| \\
&\leq \sum_{S\in\mathcal{L}(Q)} |\{(i,j) : S \in \mathcal{L}_{ij}\}| \\
&\leq d \cdot |\mathcal{L}(Q)| && (5.8) \\
&= d \cdot (N_d(Q) - K) && (5.9)
\end{aligned}
$$

Now, since Q^{SP} is the minimum weight (s, t) path in the graph, it follows that:

$$\begin{aligned}
N_d(Q^{\mathsf{SP}}) &= |\mathcal{L}(Q^{\mathsf{SP}})| + K \\
&\leq w(Q^{\mathsf{SP}}) + K \\
&\leq w(Q^*) + K \\
&\leq d \cdot (N_d(Q^*) - K) + K, \quad \text{by Equation (5.9)} \\
&\leq d \cdot N_d(Q^*).
\end{aligned}$$

□

Therefore, the number of cross-layer cuts of size d given by REROUTE_SP is at most d times the optimal reroute. Note that if the optimal new route for (s, t) eliminates every MCLC of size d, the approximation algorithm will find a new route that achieves that as well. We state this observation as the following corollary.

Corollary 5.5 REROUTE_SP(s, t) *will return a new route for* (s, t) *that increases the size of MCLC of the layered network, if such a new route exists.*

We can extend algorithm REROUTE_SP, which is based on the Dijkstra's shortest path algorithm, by using the k-shortest path algorithm [123] to successively compute the next shortest path in G'_P and keep track of the path Q with the minimum value of $|\mathcal{L}(Q)|$. The value k reflects a tradeoff between running time and quality of the solution. As we will see in Section 5.1.4, by picking a good value of k, we can obtain a lightpath routing within a much shorter time than solving the ILP without sacrificing much in solution quality.

Finally, the following theorem provides a sufficient condition for encountering the optimal route for (s, t) during the course of the successive shortest path algorithm. Specifically, if the successive shortest path algorithm returns a path with a sufficiently large weight, the algorithm can terminate right away.

Theorem 5.6 *Let* Q_i *be the* i^{th} *shortest path in the weighted graph* G'_P*, breaking ties arbitrarily. Then, for any* $i \geq 1$*, if* $w(Q_{i+1}) \geq d \cdot \min_{1 \leq j \leq i} |\mathcal{L}(Q_j)|$*, then the path* Q_{j^*}*,* *where* $j^* = \underset{1 \le j \le i}{\operatorname{argmin}} |\mathcal{L}(Q_j)|$, *is an optimal route for* (s,t).

Proof. Let $R_i = \min_{1 \le j \le i} |\mathcal{L}(Q_j)|$ be the minimum value of $|\mathcal{L}(Q)|$ among the (s,t) paths $Q_1, \ldots, Q_i$. Suppose for some i, we have $w(Q_{i+1}) \ge dR_i$. This implies all (s,t) paths Q not in $\{Q_1 \ldots, Q_i\}$ have weight $w(Q) \ge dR_i$. By Equation (5.8), $|\mathcal{L}(Q)| \ge \frac{w(Q)}{d} \ge R_i$ for all such Q. This implies Q_{j^*} is an optimal route. □

5.1.4 Simulation Results

In this section, we present our simulation results on the lightpath rerouting approach. We use the augmented NSFNET (Figure 2-3) as the physical topology, and the same set of random logical topologies in Section 2.5 as input, and run the lightpath rerouting algorithms on these instances. We will compare the reliability of the lightpath routings produced by these algorithms with the best known ILP lightpath routing formulation based on Multi-Commodity flow, presented in Section 2.4.2.

Performance of ILP-Based Rerouting

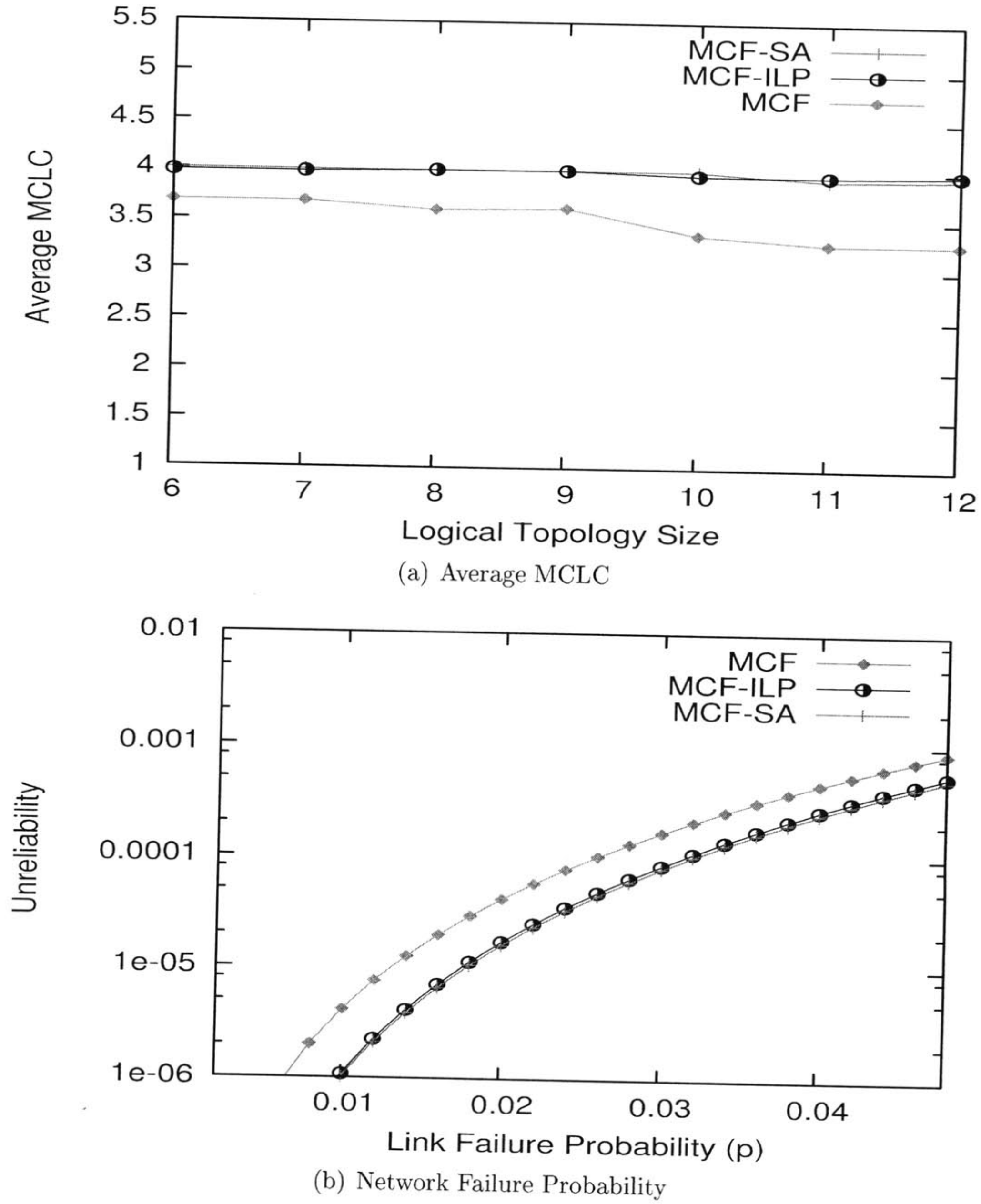
We first investigate the effectiveness of the ILP-based lightpath rerouting approach introduced in Section 5.1.2 to improve cross-layer reliability. In particular, we use the best known lightpath routing algorithm based on multi-commodity flows, $\mathsf{MCF}_{\mathsf{LF}}$, introduced in Section 2.4.2, to generate an initial set of lightpath routings. For each lightpath routing, we repeatedly solve the ILP to improve its reliability, until a local optimum is reached. We evaluate the gain in reliability achieved by this rerouting approach.

The effectiveness of the rerouting approach to improve reliability is compared with an alternative approach based on Simulated Annealing, which is a general random search technique for optimization problems. In the Simulated Annealing approach, the set of possible lightpath routings are modeled by a set of *states*, and the transition between two neighboring states represents a rerouting of a logical link. Each state is associated with a cost that reflects the reliability of the lightpath routing. In particular, a lightpath routing with higher reliability is associated with a lower cost in its corresponding state. Therefore, the state with the lowest cost corresponds to the globally optimal lightpath routing. The algorithm randomly walks over the state space, with preference towards states with lower cost, to search for the state with the lowest cost. Compared with the rerouting approach which stops at a local optimum, the Simulated Annealing approach avoids getting trapped in a local optimum by allowing non-zero probability of transitioning to neighboring states with higher costs, and thus can find the global optimum if the number of iterations is sufficiently large. Readers can refer to [65] for details about Simulated Annealing.

In this Simulated Annealing experiment, we use the constant temperature function $T(t) := 1$, and set the cost of each lightpath routing to be $N_d + 10000^{5-d}$, where d is the Min Cross Layer Cut value for the lightpath routing and N_d is the number of cross-layer cuts with size d. Therefore, the cost of a lightpath routing is smaller if it is lexicographically smaller. The Simulated Annealing algorithm starts with the same set of initial lightpath routings generated by $\mathsf{MCF_{LF}}$, and iterates until no better solution is found for 50000 iterations. The best lightpath routing encountered is returned as the output.

Figure 5-4(a) illustrates the average MCLC of the lightpath routings generated by the rerouting and Simulated Annealing algorithms. Both algorithms are able to raise the average MCLC of the initial lightpath routings to almost 4, which is the connectivity of the logical topologies and is therefore an upper bound of the MCLC value. In other words, in terms of MCLC, both algorithms provide near-optimal performance. Figure 5-4(b) illustrates the network failure probability of the lightpath routings produced by the two algorithms in the low probability regime. Again, the amount of reliability improvement achieved by both methods are very close.

Table 5.1 shows how long it takes for the two algorithms to reach their final solution, both in terms of number of iterations and running time. Simulated Annealing requires a much larger number of iterations to converge, where each iteration requires evaluating the new cost, which involves counting the number of MCLCs and is non-trivial to compute. This accounts for the long running time of Simulated Annealing. On the other hand, even though the ILP-based algorithm solves an integer program



(a) Average MCLC

(b) Network Failure Probability

Figure 5-4: Lightpath rerouting ILP vs Simulated Annealing. MCF is the original algorithm $\mathsf{MCF}_{\mathsf{LF}}$ introduced in Section 2.4.2. MCF − ILP is the ILP-based lightpath rerouting algorithm. MCF − SA is the Simulated Annealing algorithm.

in every iteration, the number of iterations is much smaller and is therefore able to converge in a much shorter time.

Number of	Number of Iterations		Running Time (seconds)	
Logical Nodes	ILP	SA	ILP	SA
6	3.0	20677	164	7622
7	4.2	29559	257	11024
8	5.0	32418	365	12600
9	6.2	32809	525	27738
10	7.3	40591	824	15567
11	8.0	34933	1280	39325
12	8.2	35471	1104	27592

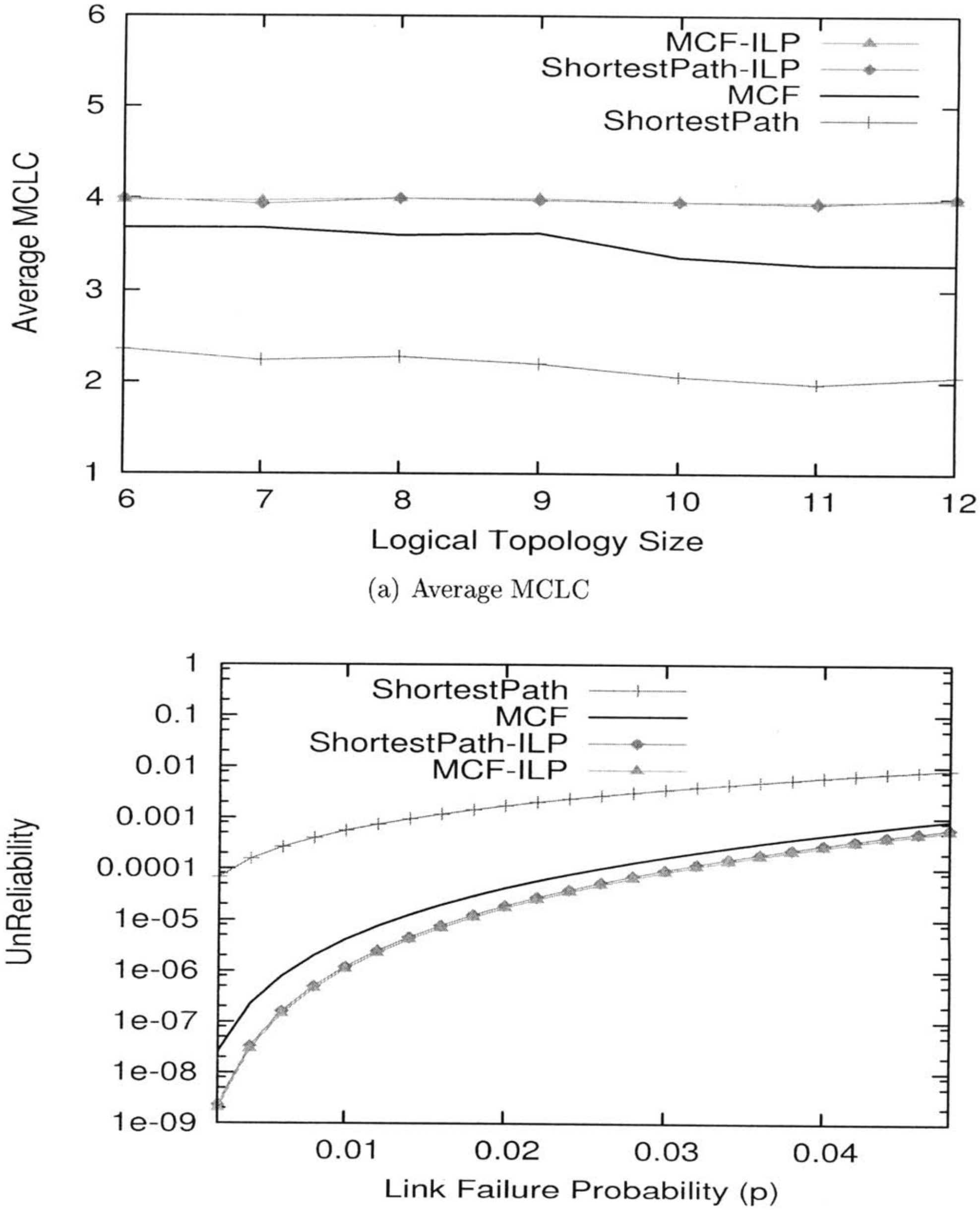
Table 5.1: Running time of the ILP and Simulated Annealing (SA) lightpath rerouting algorithms.

Robustness with Different Initial Lightpath Routings

As discussed in Section 5.1, we can repeatedly apply lightpath rerouting to any initial lightpath routing to obtain a locally optimal solution. Next, we investigate the performance of rerouting using different initial lightpath routings. We apply the ILP-based rerouting to two sets of initial lightpath routings generated by two different lightpath routing algorithms: $\mathsf{MCF}_{\mathsf{LF}}$ introduced in Section 2.4.2 and Shortest Path, which routes each lightpath with minimum number of physical hops.

Figures 5-5(a) and 5-5(b) show the average MCLC and reliability values of the two sets of lightpath routings before and after the repeated rerouting steps. Initially, the lightpath routings generated by Shortest Path have significantly lower MCLC and reliability than the ones generated by $\mathsf{MCF}_{\mathsf{LF}}$. However, the lightpath rerouting algorithm is able to improve both sets of lightpath routings to similar MCLC and reliability values. This illustrates the robustness of the lightpath rerouting approach with respect to the initial choice of lightpath routing.

Table 5.2 shows the total number of iterations and running time for the lightpath rerouting algorithm to reach the local optimum, starting with the two different sets of initial lightpath routings. As the lightpath routings generated by the shortest path algorithm generally have lower MCLC values, they require more iterations to reach the local optimum compared to the lightpath routings produced by $\mathsf{MCF}_{\mathsf{LF}}$. However, the difference in total running time is less significant. This is because the size of the rerouting ILP formulation is larger when the MCLC of the lightpath routing is large,



(a) Average MCLC

(b) Network Failure Probability

Figure 5-5: Lightpath rerouting with different initial lightpath routings.

and thus takes longer to solve. Since the lightpath routings created by the shortest path algorithm start with a lower MCLC value, most of the additional rerouting steps consist of solving the smaller ILPs to bring up the MCLC value. Therefore, these additional steps take much shorter time.

Number of Logical Nodes	Number of Iterations		Running Time (seconds)	
	MCF	SP	MCF	SP
6	3.0	7.0	164	265
7	4.2	8.9	257	314
8	5.0	10.3	365	500
9	6.2	11.6	525	745
10	7.3	14.1	824	1238
11	8.0	14.0	1280	1389
12	8.2	14.1	1104	1268

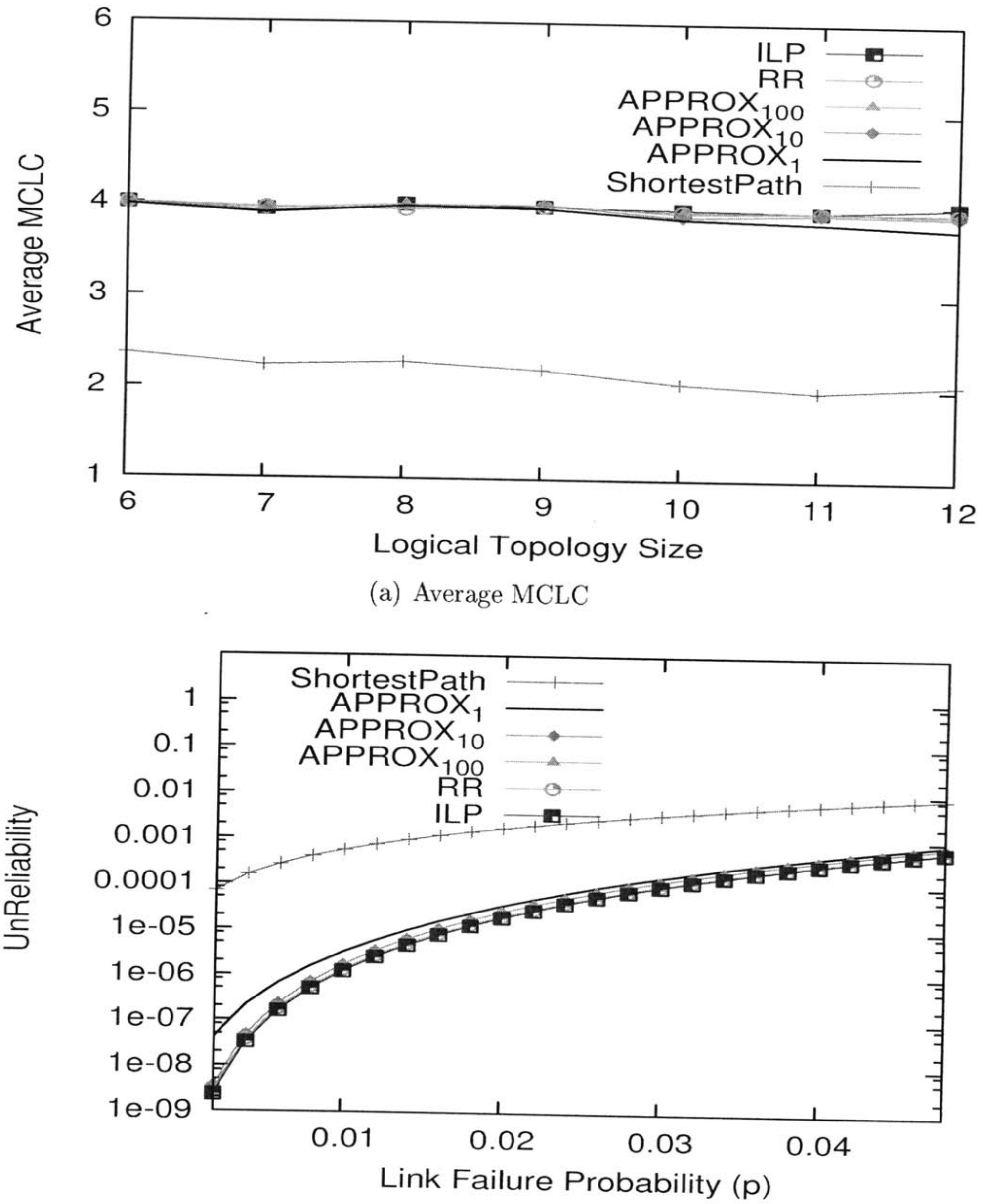
Table 5.2: Running time of iterative rerouting, with different initial lightpath routings. MCF corresponds to initial lightpath routings created by $\mathsf{MCF_{LF}}$ and SP corresponds to initial lightpath routings created by the shortest path algorithm.

Performance of Approximation Algorithm

Next, we compare the performance of the approximation algorithm introduced in Section 5.1.3 with the ILP counterpart. As discussed, the approximation algorithm is based on the k-shortest-path algorithm, where the parameter k reflects a tradeoff between running time and reliability performance. We evaluate this algorithm, $\mathsf{APPROX_k}$, with k =1, 10 and 100. In addition, we also evaluate the performance of the randomized rounding algorithm, RR, which solves the ILP REROUTE with the binary variables f_{ij} relaxed, and uses the optimal relaxed solution to construct the physical route by randomized rounding.

We use the lightpath routings generated by the Shortest Path algorithm as the initial routings. Figures 5-6(a) and 5-6(b) show the reliability performance among the algorithms. While $\mathsf{APPROX_1}$ brings in the majority of the improvement, increasing the value of k is able to further improve the reliability. In particular, when $k = 100$, the approximation algorithm performs almost as well as solving the ILP. Similarly, the randomized rounding algorithm also performs almost as well as solving the original ILP.

Table 5.3 compares the running time of each algorithm. As shown in the table, both the approximation algorithm and randomized rounding are at least several times faster than the ILP-based algorithm; and the approximation algorithm is faster overall, potentially because it does not involve solving any mathematical program



(a) Average MCLC

(b) Network Failure Probability

Figure 5-6: Lightpath rerouting: performance of approximation algorithm.

at all. This result suggests that both the approximation algorithm and randomized rounding are promising rerouting approaches to improve the reliability of lightpath routings for large networks. As we will see in Section 5.3, these algorithms continue to produce high quality solution for networks that are too large to solve the original ILP optimally.

Number of Logical Nodes	Running Time (seconds)				
	APPROX$_1$	APPROX$_{10}$	APPROX$_{100}$	RR	ILP
6	12	14	24	117	265
7	20	26	43	136	314
8	32	43	79	174	500
9	45	55	123	222	744
10	68	91	199	330	1238
11	83	104	254	397	1389
12	113	135	344	465	1268

Table 5.3: Running times of the ILP, randomized rounding and approximation algorithms.

5.2 Logical Topology Augmentation

The basic idea of *network augmentation* is to add new links to the network in order to improve the reliability of the network. Although adding new links should never hurt reliability, the marginal improvement in reliability may conceivably diminish as more links are added to the network. Thus there is a tradeoff between cost of the new links and the reliability gain from them. In this section, we will investigate the effectiveness of improving reliability of layered networks via augmentations to the logical topology.

A *logical topology augmentation*, or simply augmentation, to a layered network is defined to be a set of new logical links to be added to the network, along with their physical routes. The *Single-Link Logical Topology Augmentation Problem* involves finding the best way to augment the logical topology with a single logical link, in order to maximize the reliability improvement.

The graph augmentation problem has been extensive studied in single-layer networks. Most of the existing work [25,43,55,59,119] focuses on the problem of finding the minimum (weighted or unweighted) set of edges added to the given graph in order to satisfy a certain requirement (e.g. connectivity). Augmenting a layered network not only involves deciding which logical edges to add, but also the physical routes to take. The lightpath routing aspect of the augmentation problem makes it a much harder problem than the single-layer case.

For example, consider a network with two nodes s and t connected by n parallel edges. Suppose we would like to augment the graph so that the connectivity increases by 1. The solution in the single-layer setting would be trivial: simply add one more edge between the two nodes. However, in the multi-layer setting, the minimum number of additional logical links required to increase the MCLC depends on the underlying physical topology as well as the lightpath routing. Therefore, augmenting layered networks to improve reliability appears to be a more challenging problem.

In the following, we will study the single-link augmentation problem. We first give a characterization of the problem in Section 5.2.1, and discuss its similarity with the lightpath rerouting problem studied in Section 5.1. We next develop a similar ILP formulation and approximation algorithm in Sections 5.2.2 and 5.2.3, and present some empirical results from a case study of augmenting logical rings in Section 5.2.4. We will look into the structure of the augmentation problem in Section 5.2.5, and derive a lower bound on the minimum number of logical links required to increase the MCLC of the network. The lower bound will be used in Section 5.2.6 to evaluate the augmentation algorithm based on repeated single-link augmentations.

5.2.1 Effects of a Single-Link Augmentation

Given a lightpath routing for the physical topology $G_P = (V_P, E_P)$ and logical topology $G_L = (V_L, E_L)$, the Single-Link Logical Topology Augmentation problem is to find two logical nodes $s, t \in V_L$, and a (s, t) path in G_P, such that the reliability of the network is maximized by augmenting the network with the new logical link using the specified physical path. Similar to the rerouting problem, such a logical link should maximize the reduction in the number of MCLCs. In fact, since rerouting a logical link can be considered as removing an existing logical link from the logical topology, and then augmenting the logical topology with a new link between the two nodes. It is thus not surprising that the characterizations for the single-link augmentation problem is similar to the lightpath rerouting problem. However, unlike rerouting, augmenting the logical topology with a new link never converts a non-cut into a cross-layer cut. Therefore, in augmentation we only need to consider the effect of the new logical link on the existing cross-layer cuts.

Suppose that an initial lightpath routing is given for the physical topology $G_P =$

(V_P, E_P) and the logical topology $G_L = (V_L, E_L)$. Let d be the size of the MCLC under the initial routing. Let G_L^S be the logical residual graph for any cross-layer cut S, that is, the logical subgraph in which the logical links do not use any physical links in S. The following theorem characterizes the effect of a single-link augmentation:

Theorem 5.7 *Let S be a cross-layer cut for a lightpath routing. Augmenting the network with a new logical link (s, t) over physical route P converts a cross-layer cut S into a non-cut if and only if:*

1. *S is a 2-way cross-layer cut.*

2. *s and t are disconnected in the residual graph for S.*

3. *P does not use any physical links in S.*

Proof. The proof is the same as Theorem 5.1. The new logical link will make the residual graph connected if and only if the above conditions are true. □

Note that the conditions in Theorem 5.7 are the same as Theorem 5.1. Therefore, the algorithms presented in Sections 5.1.2 and 5.1.3 are mostly applicable here.

5.2.2 ILP for Single-Link Logical Topology Augmentation

The ILP for the single-link logical topology augmentation problem is similar to the formulation in Section 5.1.2, and can be interpreted as a path selection problem on the auxiliary graph $G_P' = (V_P', E_P')$, where $V_P' = V_P \cup \{u, v\}$ and $E_P' = E_P \cup \{(u, s), (s, v) : \forall s \in V_L\}$, as shown in Figure 5-3.

Let d be the size of the MCLC and $\mathcal{C}_d$ be the set of 2-way cross-layer cuts of size d in the given lightpath routing. We first define the following variables and parameters:

1. Variables:

   - $\{g_{st} : (s, t) \in V_L \times V_L\}$: 1 if logical link (s, t) is added to the network, and 0 otherwise.

- $\{f_{ij} : (i,j) \in E'_P\}$: Flow variables describing a path in G'_P from node u to node v.
- $\{y^c : c \in \mathcal{C}_d\}$: 1 if the cross-layer cut c is converted into a non-cut by the augmentation, and 0 otherwise.

2. Parameters:

- $\{h^c_{st} : c \in \mathcal{C}_d, (s,t) \in E_L\}$: 1 if logical nodes s and t are disconnected by the 2-way cut c, and 0 otherwise.
- $\{l^c_{ij} : \forall c \in \mathcal{C}_d, (i,j) \in E_P\}$: 1 if physical link (i,j) is in the set of physical links c, and 0 otherwise.

The logical augmentation problem can then be formulated as the following ILP:

$$\textsf{AUGMENT}: \quad \text{Maximize} \sum_{c \in \mathcal{C}_d} y^c, \quad \text{subject to:}$$

$$g_{st} \leq (f_{us} + f_{tv})/2, \quad \forall (s,t) \in V_L \times V_L \tag{5.10}$$

$$y^c \leq \sum_{(s,t) \in V_L \times V_L} h^c_{st} g_{st}, \quad \forall c \in \mathcal{C}_d \tag{5.11}$$

$$y^c \leq 1 - l^c_{ij} f_{ij}, \quad \forall (i,j) \in E_P, \forall c \in \mathcal{C}_d \tag{5.12}$$

$$\{(i,j) : f_{ij} = 1\} \text{ forms an } (u,v)\text{-path in } G'_P \tag{5.13}$$

$$f_{ij}, g_{st} \in \{0,1\}, 0 \leq y^c \leq 1$$

In a feasible solution to the formulation, the variables f_{ij} represent a path $u \to s \rightsquigarrow t \to v$, as described by Constraint (5.13). This corresponds to the new logical link to be added to the network, along with its physical route. Constraint (5.10) ensures that $g_{st} = 1$ if and only if (s,t) is the new logical link selected. Constraints (5.11) and (5.12) describe the conditions in Theorem 5.7. The variable y^c describes whether the cross-layer cut c is converted into non-cut by the augmentation. Therefore, the ILP maximizes the number of such conversions, which translates to maximizing the improvement in reliability.

5.2.3 An Approximation Algorithm For Logical Topology Augmentation

One can also design an approximation algorithm similar to REROUTE_SP introduced in Section 5.1.3 for the single-link logical topology augmentation problem. We will again focus on the following question: Given a layered network, and a new logical link (s, t), find the physical route for (s, t) such that the resulting number of cross-layer cuts of size d is minimized. We can then apply the algorithm for this problem for every possible pair of logical nodes s and t, to find out the new logical link that would result in the maximum reliability improvement.

Let d be the size of the MCLC of the layered network and $\mathcal{C}_d^{st}$ be the set of 2-way cross-layer cuts of size d that separate the logical nodes s and t. Then by Theorem 5.7, the set $\mathcal{L}_{ij} = \{S \in \mathcal{C}_d^{st} : (i, j) \in S\}$ represents the sets in $\mathcal{C}_d^{st}$ that will remain to be cross-layer cuts if the physical link (i, j) is used by the (s, t) path Q. We can then develop an approximation algorithm for the augmentation problem similar to REROUTE_SP:

Algorithm 5 AUGMENT_SP(s, t)

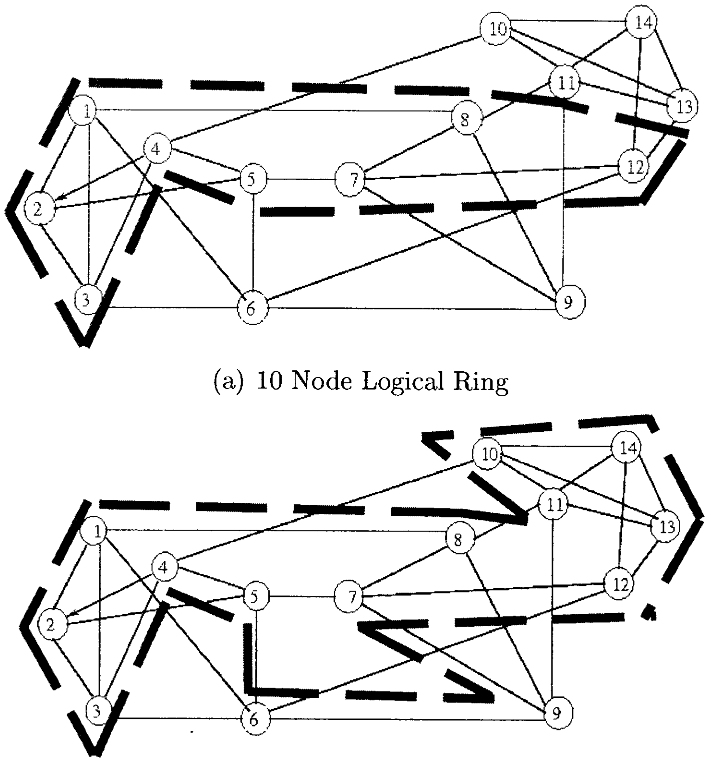
```
Construct a weighted graph on $G_P = (V_P, E_P)$, where each edge $(i, j)$ is assigned
    with weight $w(i, j) = |\mathcal{L}_{ij}|$.
Run Dijkstra's algorithm to find the shortest $(s, t)$-path in the weighted graph.
```

Since each cross-layer cut S in $\mathcal{C}_d^{st}$ has size d, there are exactly d physical links (i, j) such that $S \in \mathcal{L}_{ij}$. As a result, AUGMENT_SP is a d-approximation algorithm, with the same proof as Theorem 5.4.

5.2.4 A Case Study: Augmenting a Logical Ring

In this section, we consider augmenting logical rings of different sizes to study the reliability improvement by the augmentation approach. We start with a 10-node and 14-node logical rings on the augmented NFSNET, as shown in Figure 5-7, and run the single link augmentation algorithm repeatedly.

The cross-layer reliability of the networks after each augmentation step is shown



(a) 10 Node Logical Ring

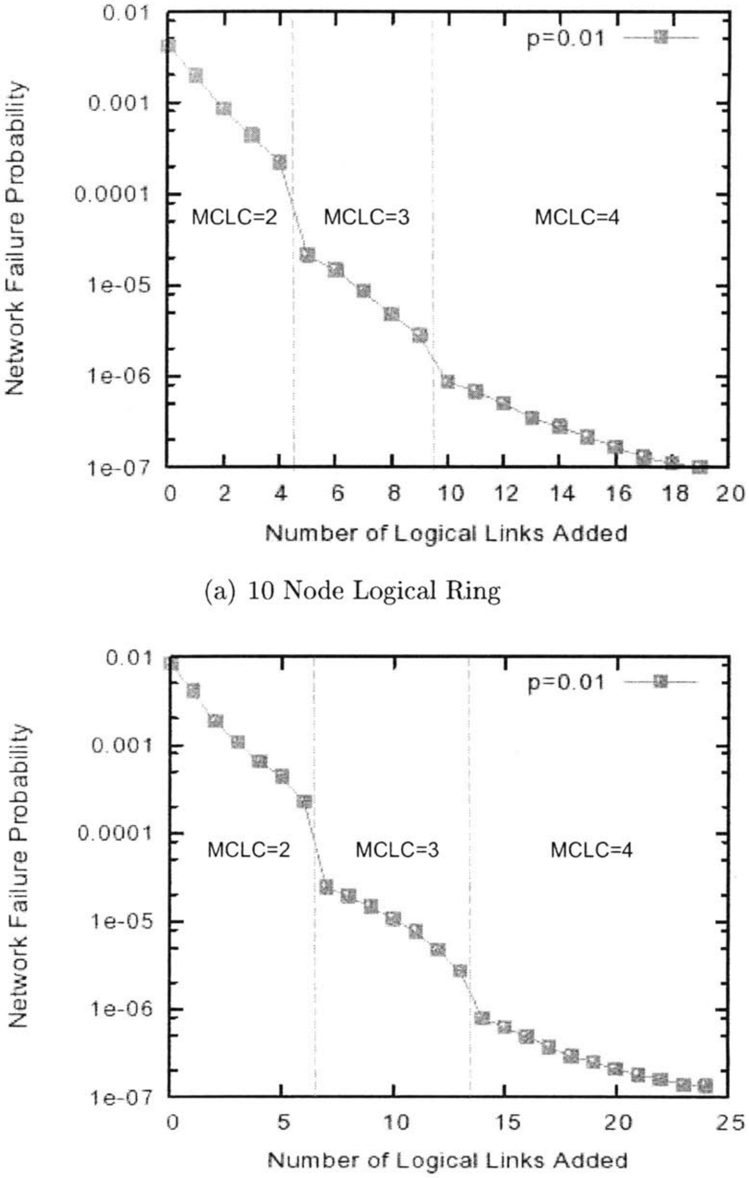
(b) 14 Node Logical Ring

Figure 5-7: Logical rings on extended NSFNET.

in Figure 5-8. With link failure probability $p = 0.01$, the unreliability declines as we add more logical links to the rings. The key observation from these figures is that the improvement in reliability is most prominent when the augmentation increases the MCLC of the network. This further validates our approach to maximize the MCLC as the primary objective. In the case where the additional link does not cause an MCLC increase, the marginal reliability improvement decreases with the current MCLC value. This means that augmentation is most effective when MCLC is low.

5.2.5 Minimum Augmenting Edge Set

Based on the observation from the case study, the *Minimum Augmenting Edge Set*, defined to be the smallest set of new logical links required to increase the MCLC of the layered network, is of particular interest. Clearly, the MCLC value for a layered network is upper bounded by the the logical connectivity. Therefore, given a layered network with MCLC value d, the number of new logical links needed to increase the



(a) 10 Node Logical Ring

(b) 14 Node Logical Ring

Figure 5-8: Impact on reliability by augmenting logical rings.

MCLC value is at least the number of edges required to augment the logical topology to connectivity $d + 1$. This gives a simple lower bound on the size of the minimum augmenting edge set.

In the case of logical rings of size n, this means at least $\lceil \frac{n}{2} \rceil$ logical links are required to increase the MCLC, which happens to be tight for the results in Figure 5-8. In other words, augmenting the network incrementally using the single-link augmentation ILP performs optimally in this particular case.

In general, however, a logical topology with high connectivity can still have low MCLC when embedded in a physical network, and this simple lower bound will not be useful. In the next section, we present a method to establish a tighter lower bound.

Lower Bound on Minimum Augmenting Edge Set

We can develop a tighter lower bound on the size of the minimum augmentation edge set by taking the structure of lightpath routing into account. Suppose we are given the physical topology $G_P = (V_P, E_P)$, logical topology $G_L = (V_L, E_L)$ and the lightpath routing, we start with a few definitions.

Definition 5.1 *Given the lightpath routing, a set of logical links L is* **covered** *by a set of physical links C if all of the links in L use at least a physical link in C.*

Definition 5.2 *A subset of logical nodes $S \subset V_L$ is d-***protected** *if and only if the logical cut set $\delta(S)$ is not covered by any set of d physical links. In other words, given any d-physical link failure, at least one of the logical links in $\delta(S)$ survives.*

Definition 5.3 *The d-***deficit** *$\lambda_d(S)$ for a subset of logical nodes $S \subset V_L$ is the minimum number of new logical links in $(S, V_L - S)$ that needs be added in order to make S d-protected. If S cannot be made d-protected (because the connectivity of the physical topology is less than d), $\lambda_d(S)$ is defined to be ∞.*

The following theorem relates the d-protectedness of the logical node sets to the MCLC of the layered network.

Theorem 5.8 *The MCLC of a layered network is at least $d + 1$ if and only if S is d-protected for all $S \subset V_L$.*

Proof. Suppose there exists a set of logical nodes $S \subset V_L$ that is not d-protected. Then there exists a set of d physical links that cover all logical links in $\delta(S)$. As a result, failure of this set of physical links will disconnect S from the rest of the logical topology, implying that the MCLC is at most d.

On the other hand, suppose the node set S is d-protected for all $S \subset V_L$. Then after removing any d physical links from the layered network, at least one logical link in $\delta(S)$ survives for any $S \subset V_L$, which implies that the MCLC is at least $d+1$. □

The next theorem provides the framework in establishing the lower bound on the size of the minimum augmenting edge set for a lightpath routing.

Theorem 5.9 *Given a layered network, let d be the MCLC value. The minimum augmenting edge set for the layered network is at least $\frac{1}{2} \sum\limits_{V_L^i \in \mathcal{T}} \lambda_d(V_L^i)$, for any partition $\mathcal{T} = \{V_L^1, \ldots, V_L^k\}$ of the logical node set V_L.*

Proof. Any augmenting edge set Y that increases the MCLC of the network to $d+1$ must make V_L^i d-protected, by Theorem 5.8. By definition of λ_d, for all i, such an augmenting edge set must contain $\lambda_d(V_L^i)$ logical links with one end point in V_L^i. This implies that Y must contain at least $\frac{1}{2} \sum\limits_{V_L^i \in \mathcal{T}} \lambda_d(V_L^i)$ logical links. □

Theorem 5.9 suggests that we can choose any partition of V_L and establish a lower bound by computing the deficit $\lambda_d(V_L^i)$ for each component in the partition. We will discuss how the deficit can be computed in Appendix 5.5. In the rest of this section, we will discuss how to choose a good partition of V_L to establish a meaningful lower bound.

Definition 5.4 *Two logical nodes x and y are d-**connected** if they stay logically connected to each other under any set of $d-1$ physical failures.*

The following theorem shows that d-connectedness is a transitive relation.

Theorem 5.10 *Given logical nodes x, y, z in a layered network, if x is d-connected to y, and y is d-connected to z, then x is d-connected to z.*

Proof. Suppose x is not d-connected to z. Then there exists a set of $d-1$ physical links C whose removal will disconnect nodes x and z. Therefore, the node y will be disconnected from either x or z on the removal of C, implying that either x, y are not d-connected or y, z are not d-connected, which is a contradiction. □

Given any partition $\mathcal{T} = \{V_L^1, \ldots, V_L^k\}$ of V_L, if there exist x and y that are d-connected to each other such that they belong to different components V_L^i and V_L^j, then $\lambda_d(V_L^i) = \lambda_d(V_L^j) = 0$. As a result $\lambda_d(V_L^i \cup V_L^j) \geq \lambda_d(V_L^i) + \lambda_d(V_L^j)$. In other words, the sum $\sum\limits_{V_L^i \in \mathcal{T}} \lambda_d(V_L^i)$ in Theorem 5.9 will not decrease if the components V_L^i and V_L^j are merged. This motivates the following procedure:

Algorithm 6 MERGE_COMPONENT(s, t)

```
1: Create an initial partition for $V_L$: $\mathcal{T} := \left\{V_L^1, \ldots, V_L^{|V_L|}\right\}$, where each component
   $V_L^i$ contains a single logical node.
2: while $\exists x \in V_L^i, y \in V_L^j, i \neq j$, such that $x$ and $y$ are $d$-connected, do:
     Replace $V_L^i, V_L^j$ in $\mathcal{T}$ by $V_L^i \cup V_L^j$.
3: Return $\mathcal{T}$.
```

At the end of the procedure, each component V_L^i in the partition $\mathcal{T}$ output by MERGE_COMPONENT contains nodes that are d-connected to one another, and nodes across different components are not d-connected. Therefore, this partitioning exposes components among which logical links need to be added.

5.2.6 Simulation Results

In Section 5.2.2, we presented an ILP formulation for the single-link augmentation problem to maximize the reliability improvement. One can repeatedly apply the algorithm to incrementally augment the network to construct an augmenting edge set. In this section, we will compare the solution provided by this approach with the lower bound given in Section 5.2.5.

Using the augmented NSFNET (Figure 2-3) as the physical topology and the same set of 350 random logical topologies as in Section 5.1.4, we considered lightpath routings with MCLC values 3, and studied the number of new logical links needed by the algorithm to raise the MCLC values to 4. This number is compared to the lower bound given by Theorem 5.9. Note that the simple lower bound introduced at the beginning of Section 5.2.5 based on logical connectivity would not be helpful in this case, since the connectivity of the logical topologies is already 4.

The number of new logical links needed by the algorithm, as well as the lower bound given by Theorem 5.9, are shown in Figure 5-9. In 330 of the 350 instances, the number of logical links required by the algorithm is able to meet the lower bound, whereas in the other 20 instances the number is one larger than the lower bound. This suggests that the incremental augmentation approach is able to come up with an optimal or near-optimal augmenting edge set in each case. In addition, the result shows that Theorem 5.9 gives us a good lower bound that can be used for evaluating augmentation algorithms.

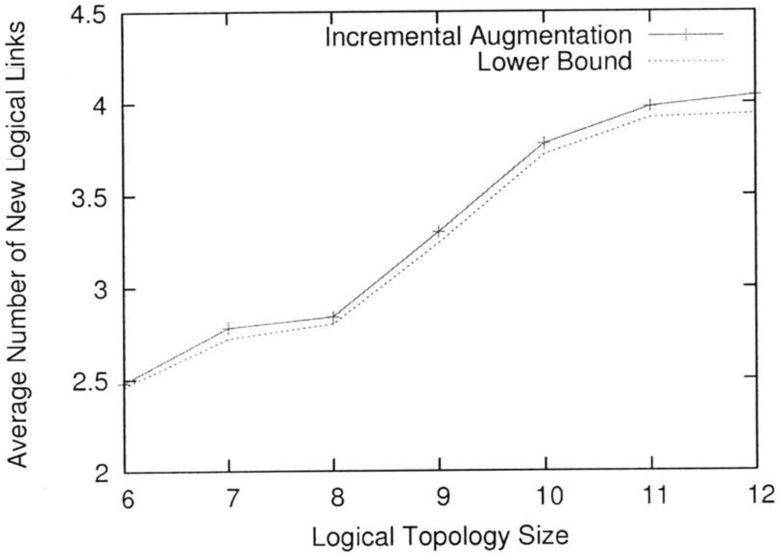


Figure 5-9: Size of augmenting edge set generated by incremental single-link augmentation vs lower bound.

Finally, we study the marginal benefit of augmenting the logical topology, using the lightpaths routings produced by the rerouting method in Section 5.1.4 as the baseline. Figure 5-10 shows the improvement in reliability by augmenting the network with different number of logical links. As the starting lightpath routings already achieve the maximum possible MCLC value, the improvement shown in the figure is due to the reduction in the number of MCLCs. Even though the marginal improvement in reliability diminishes with more logical links added to the network, overall, the reliability of the network can be further improved by augmentation.

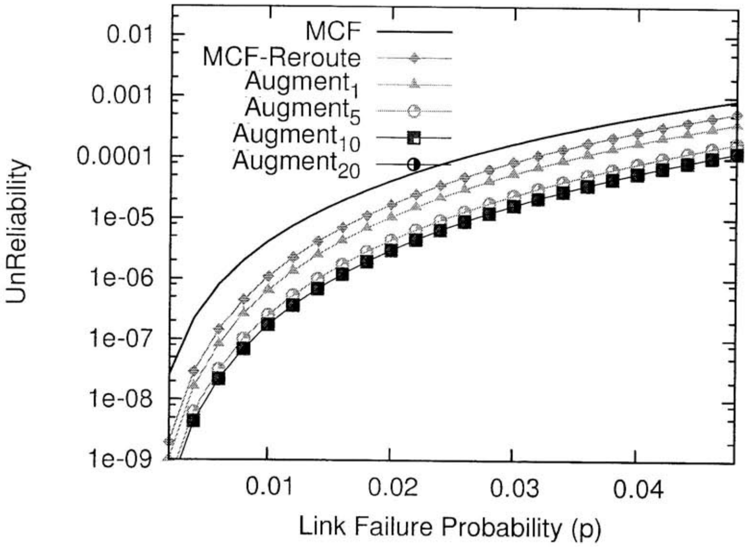


Figure 5-10: Improving reliability via augmentation.

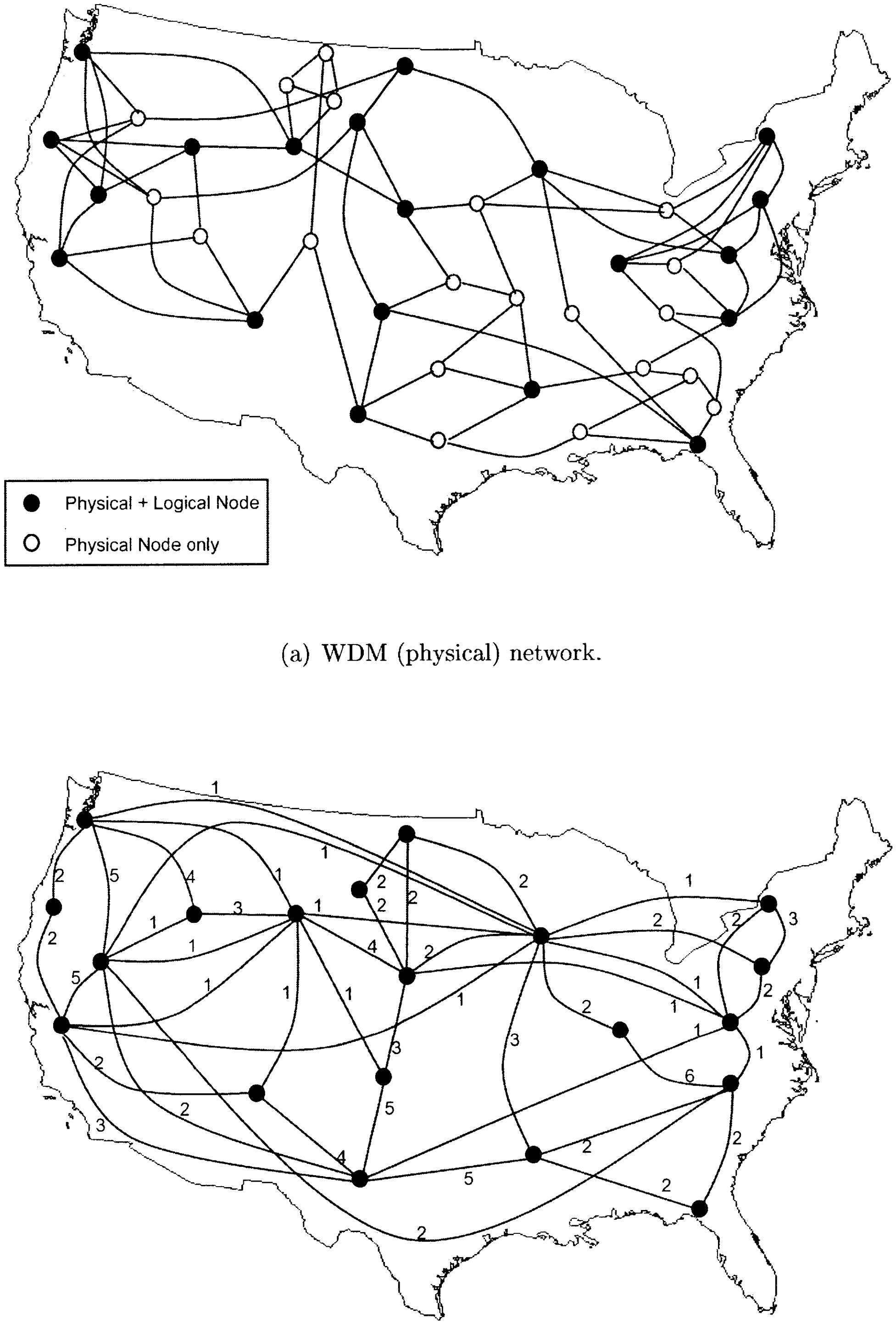
5.3 Case Study: A Real-World IP-Over-WDM Network

Most of the simulations presented in this thesis are on the 14-node augmented NSFNET as the physical topology. In this section, we will study the performance of various algorithms on a large layered network based on a real-world IP-over-WDM network. The physical and logical topologies, shown in Figure 5-11, are constructed based on the network maps available from Qwest Communications [1].

The study on networks of larger size allows us to reevaluate the performance of the lightpath algorithms, both in terms of scalability and solution quality. In this study, we have attempted to run the various lightpath routing algorithms introduced throughout the thesis, including:

1. SURVIVE: The existing survivable lightpath routing algorithm introduced in [76], used as the benchmark for comparing with the algorithms introduced in this thesis. The lightpath routing is computed using randomized rounding (Section 2.4.3) on the optimal solution of the linear relaxation.

2. $\text{MCF}_{\text{MinCut}}$: The simple multi-commodity flow formulation introduced in Sec-



(a) WDM (physical) network.

(b) IP/MPLS (logical) network. The numbers indicate the number of parallel logical links between the logical nodes.

Figure 5-11: Physical and logical topologies.

tion 2.4.2. The lightpath routing is computed using randomized rounding on the optimal solution of the linear relaxation.

3. $\mathsf{MCF}_{\mathsf{LF}}$: The enhanced multi-commodity flow formulation introduced in Section 2.4.2, where each constraints captures the impact of a fiber failure on each logical cut. The lightpath routing is computed using randomized rounding on the optimal solution of the linear relaxation.

4. $\mathsf{REROUTE}_{\mathsf{ILP}}$: The iterative lightpath rerouting algorithm, based on the ILP presented in Section 5.1.2.

5. $\mathsf{REROUTE}_{\mathsf{RR}}$: The iterative lightpath rerouting algorithm, based on the ILP presented in Section 5.1.2, with the variables f_{ij} relaxed. The physical route is obtained by choosing the best solution out of 1000 iterations of randomized rounding on the optimal fractional solution to f_{ij},

6. $\mathsf{REROUTE}_{\mathsf{Approx}}$: The iterative lightpath rerouting algorithm, based on the k-shortest path algorithm presented in Section 5.1.3, where k is set to 5000 in our experiment.

7. $\mathsf{AUGMENT}_{\mathsf{ILP}}$: The logical topology augmentation algorithm, based on the ILP presented in Section 5.2.2.

8. $\mathsf{AUGMENT}_{\mathsf{Approx}}$: The logical topology augmentation algorithm, based on the k-shortest path algorithm presented in Section 5.2.3, where k is set to 5000 in our experiment.

Table 5.4 summarizes the results of the lightpath routing algorithms. In general, algorithms that solve ILPs (such as $\mathsf{REROUTE}_{\mathsf{ILP}}$, and $\mathsf{AUGMENT}_{\mathsf{ILP}}$) or large linear programs (such as $\mathsf{MCF}_{\mathsf{LF}}$) are no longer feasible, due to the large memory requirement of the ILP and LP solvers. This limitation of ILP-based solution justifies the design of more scalable methods, such as the randomized rounding algorithm $\mathsf{REROUTE}_{\mathsf{RR}}$; as well as the approximation algorithms $\mathsf{REROUTE}_{\mathsf{Approx}}$ and $\mathsf{AUGMENT}_{\mathsf{Approx}}$. The approximation algorithms, which are based on the successive shortest path algorithm, run in polynomial time and require a much smaller memory footprint than solving the ILP, and are therefore able to finish successfully for networks of this scale.

Algorithm	Terminates Successfully ?
SURVIVE	Yes
MCF$_{\text{MinCut}}$	Yes
MCF$_{\text{LF}}$	No
REROUTE$_{\text{ILP}}$	No
REROUTE$_{\text{RR}}$	Yes
REROUTE$_{\text{Approx}}$	Yes
AUGMENT$_{\text{ILP}}$	No
AUGMENT$_{\text{Approx}}$	Yes

Table 5.4: Scalability comparisons among different lightpath routing algorithms.

We next compare the quality of the lightpath routings produced by the algorithms SURVIVE, MCF$_{\text{MinCut}}$, REROUTE$_{\text{RR}}$, REROUTE$_{\text{Approx}}$ and AUGMENT$_{\text{Approx}}$ (with different number of new logical links). The MCLC values and the number of MCLCs of the lightpath routings generated by each algorithm are shown in Table 5.5. These numbers are compared against the lower bound, which is computed by counting the number of minimum sized physical fiber sets whose removal will *physically* disconnect some logical nodes. These sets of hysical links are cross-layer cuts regardless of the lightpath routing, and therefore will provide a lower bound on the number of MCLCs.

It was observed in Section 2.5 that the survivability performance of the multi-commodity flow formulation MCF$_{\text{MinCut}}$ declines as the network size increases. In this case, the MCLC value of the lightpath routing produced by MCF$_{\text{MinCut}}$ is no better than SURVIVE, although by spreading the logical links over different physical fibers, the algorithm manages to reduce the number of logical cuts that are covered by a 2-fiber failure. On the other hand, the rerouting algorithms REROUTE$_{\text{RR}}$ and REROUTE$_{\text{Approx}}$ continue to be able to improve the MCLC to the maximum possible value of 4 (limited by the physical connectivity). Augmenting the logical topology can further improve the reliability of the layered network by reducing the number of MCLCs, though the incremental effect declines as more logical links are added to the network. The number of MCLCs hits the lower bound when the logical topology is augmented with 9 additional logical links.

Figure 5-12 compares the algorithms in terms of the cross-layer reliability in the low failure probability regime. Consistent with Table 5.5, the iterative algorithms presented in this chapter achieve significantly higher reliability than the joint lightpath routing algorithms. In particular, the majority of the improvement is achieved by the lightpath rerouting approach, especially by the approximation algorithm $\mathsf{REROUTE}_{\mathsf{Approx}}$. Therefore, even if adding new logical links is not an option, the lightpath rerouting method allows us to obtain a lightpath routing that is close to optimal. In summary, the approximation algorithms introduced in Sections 5.1.3 and 5.2.3 provide a good tradeoff between scalability and solution quality.

Algorithm	MCLC	Number of MCLCs
$\mathsf{SURVIVE}$	2	26
$\mathsf{MCF}_{\mathsf{MinCut}}$	2	5
$\mathsf{REROUTE}_{\mathsf{RR}}$	4	458
$\mathsf{REROUTE}_{\mathsf{Approx}}$	4	216
$\mathsf{AUGMENT}_{\mathsf{Approx-1}}$	4	84
$\mathsf{AUGMENT}_{\mathsf{Approx-2}}$	4	49
$\mathsf{AUGMENT}_{\mathsf{Approx-3}}$	4	34
$\mathsf{AUGMENT}_{\mathsf{Approx-4}}$	4	29
$\mathsf{AUGMENT}_{\mathsf{Approx-5}}$	4	25
$\mathsf{AUGMENT}_{\mathsf{Approx-6}}$	4	23
$\mathsf{AUGMENT}_{\mathsf{Approx-7}}$	4	22
$\mathsf{AUGMENT}_{\mathsf{Approx-8}}$	4	21
$\mathsf{AUGMENT}_{\mathsf{Approx-9}}$	4	20
Lower Bound	**4**	**20**

Table 5.5: MCLC values and MCLC counts of different lightpath routings. The lightpath routing on a logical topology augmented with k new logical links is denoted by $\mathsf{AUGMENT}_{\mathsf{Approx-k}}$.

5.4 Conclusion

In this chapter, we propose two methods to improve the reliability of a layered network in the low failure probability regime. The main idea behind these methods is to maximize the size of the MCLC, as well as minimize the number of MCLCs via

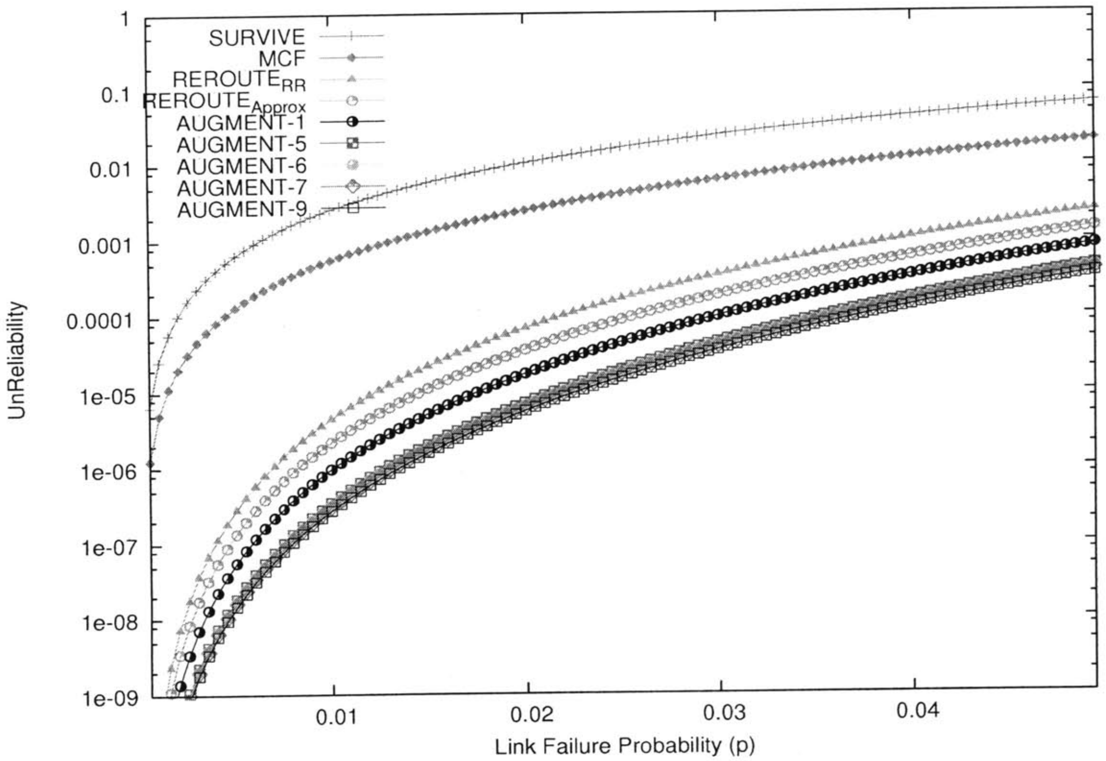


Figure 5-12: Unreliability of different lightpath routings.

iterative local changes to the layered network. In the *lightpath rerouting* method, each iterative step involves replacing the physical route of an existing logical link by a new route that results in a smaller number of MCLCs. In the *logical topology augmentation* method, each iteration augments logical topology with a new link that eliminates the maximum number of MCLCs. By applying the methods iteratively to a layered network, we can obtain a locally optimal lightpath routing in the low failure probability regime.

For both the rerouting and augmentation problems, we develop an ILP, as well as a polynomial time approximation algorithm, to compute a (near-)optimal solution in each iteration. Simulation results show that through such iterative incremental improvements, we can obtain a lightpath routing with significantly higher reliability than any existing lightpath routing algorithms, including the algorithms introduced in Chapter 2.

The iterative approach introduced in this chapter is also more scalable in general compared with the conventional lightpath routing algorithms, which compute the physical route for all logical links jointly. By considering only local changes one logical link at a time, the reliability optimization problem is broken down into smaller and manageable subproblems, which can then be efficiently solved by the approximation algorithm. This provides a viable approach to the design of reliable layered networks of large scale in the real world.

5.5 Chapter Appendix: Computing Deficit of a Logical Node Set

In Section 5.2.5, we define the *d-deficit* $\lambda_d(S)$ of a logical nodes set S to be the minimum number of logical links that need to be added to make S d-protected, given a layered network with MCLC d. In this section, we discuss how this value can be computed.

First note that sometimes it is impossible to make the node set S d-protected. For example, if there are only d physical fibers that connect S to other physical nodes, the failure of these d links will disconnect all logical links that connect S to $V_L - S$. In that case, $\lambda_d(S)$ is defined to be ∞. In the rest of the section, we assume that the physical topology is $d+1$ connected, so that it is possible to make the node set S d-protected.

We will present an ILP that computes the smallest set of new logical links to make the node set S d-protected. The ILP relies on the following result:

Theorem 5.11 $\lambda_d(S) \leq d+1$.

Proof. Pick $x \in S$ and $y \in V_L - S$. Since the physical topology is $(d+1)$-connected, there exists $d+1$ physically disjoint paths between x and y. Therefore, if we add $d+1$ copies of new logical links (x, y), each taking on one of the physically disjoint paths, at least one of the links would survive against any d-fiber failure. Therefore, $\lambda_d(S) \leq d+1$. □

As a result of the theorem, we can formulate an ILP to select up to $d+1$ paths between S and $V_L - S$, such that for any cross-layer cut C of size d, at least one of the paths do not use any fibers in C. Given the physical topology $G_P = (V_P, E_P)$, we construct an auxiliary graph $G_P^{'} = (V_P^{'}, E_P^{'})$, where $V_P^{'} = V_P \cup \{u, v\}$ and $E_P^{'} = E_P \cup \{(u, x) : x \in S\} \cup \{(x, v) : x \in V_L - S\}$, as shown in Figure 5-13. In the auxiliary graph, a new source node u and a sink node v are added, and the source node is connected to all nodes in S, and the sink node v is connected to all nodes in $V_L - S$. As a result, any (u, v) path in $G_P^{'}$ corresponds to a logical link from S to $V_L - S$ as well as its physical route.

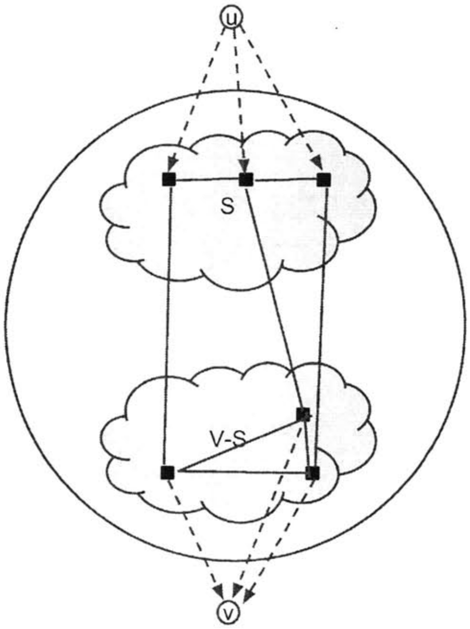


Figure 5-13: Auxiliary graph $G_P^{'}$ for the ILP. Nodes u and v are the new source and sink nodes, and the dashed lines are the new edges.

We first define the following variables and parameters for the ILP.

1. Parameters:

   - S: the logical node set for which $\lambda_d(S)$ is to be computed.
   - $\mathcal{C}_d^S$: cross-layer cuts of size d that cover all logical links in $\delta(S, V_L - S)$ in the original logical topology G_L.
   - $\{l_{ij}^c : \forall c \in \mathcal{C}_d^S, (i, j) \in E_P\}$: 1 if physical link (i, j) is in fiber set c, and 0 otherwise.

2. Variables:

   - $\{f_{ij}^k : (i,j) \in E_P', 1 \le k \le d+1\}$: Flow variables describing the k^{th} path in G_P' from node u to node v.
   - $\{y_k^c : c \in \mathcal{C}_d^S, 1 \le k \le d+1\}$: 1 if the k^{th} path uses any fiber in cross-layer cut c, and 0 otherwise.

The deficit of the node set S can be computed by the formulation below:

Minimize ρ, subject to:

$$\rho = \sum_{1 \le k \le d+1} \sum_{x \in S} f_{ux}^k \tag{5.14}$$

$$y_k^c \ge l_{ij}^c f_{ij}^k, \quad \forall (i,j) \in E_P, c \in \mathcal{C}_d^S, 1 \le k \le d+1 \tag{5.15}$$

$$\sum_{1 \le k \le d+1} y_k^c \le \rho - 1, \quad \forall c \in \mathcal{C}_d^S \tag{5.16}$$

$$\{(i,j) : f_{ij}^k = 1\} \text{ is all 0, or forms an } (u,v)\text{-path in } G', \forall 1 \le k \le d+1$$

$$f_{ij}^k \in \{0,1\}, 0 \le y_k^c \le 1$$

The formulation selects up to $d+1$ paths from u to v. Each path represents a new logical link that will be added to the logical topology. Constraint (5.14) counts the number of new logical links selected. The variable y_k^c indicates whether the k^{th} logical link, if selected, will be disconnected by cross-layer cut c, by Constraint (5.15). Constraint (5.16) then ensures that for any cross-layer cut c, at least one of the new logical links will survive its failure. As a result, the solution given by the formulation will make the node set S d-protected, and the optimal value equals the value of $\lambda_d(S)$.

Chapter 6

Conclusion and Future Work

In this thesis, we consider a layered network model where the upper-layer logical links share the lower-layer physical fibers via lightpath routing. As such, a single physical failure will cause multiple logical links to fail in a correlated manner. This phenomenon introduces new challenges in defining, measuring and optimizing survivability in the layered setting. This thesis investigates the new issues that arise under this model, in an attempt to develop useful insights in survivable layered network design.

We start with an investigation of the fundamental properties of layered networks, and show that basic connectivity structures, such as cuts, disjoint paths and spanning trees, exhibit fundamentally different characteristics from their single-layer counterparts. This necessitates the pursuit of new survivability metrics that properly quantify the resilience of the network against physical failures. To this end, we define a new metric, the Min Cross Layer Cut (MCLC), to be our primary cross-layer metric and develop algorithms to design layered networks with high MCLC values.

We next extend our study to a setting where physical link failures are modelled as random events. Under this model, we study the *cross-layer reliability* of layered networks, defined to be the probability that the logical topology stays connected under the random physical failures. The key to this study is the failure polynomial, which expresses the cross-layer reliability of the network as a polynomial in the physical link failure probability. The coefficients of the polynomial contain important structural information about the layered network. By exploiting the structures of cross-layer cuts in a layered network, we develop an efficient algorithm to estimate the cross-layer reliability.

Through the study of the failure polynomial, we also develop important insight into the connection between the link failure probability, the cross-layer reliability and the structure of a layered network. For the cases where the link failure probability is sufficiently low or sufficiently high, we have characterized the optimality conditions for lightpath routings, and developed bounds on the failure probability regimes where these conditions apply. This result also leads to a non-trivial sufficient condition for uniformly optimal lightpath routing.

Based on these insights, we develop new algorithms to design layered networks that are optimized for the low failure probability regime. Based on the ideas of iterative rerouting and augmentation, these algorithms are able to achieve locally optimal solutions. Our simulation results show that lightpath routings produced by these methods are significantly more reliable than the lightpath routings produced by existing algorithms, and are more scalable to large networks.

Throughout the thesis, we have considered the connectedness of the logical topology as the survivability requirement, and defined metrics, such as MCLC, based on this. One natural extension to our study is to consider different survivability requirements. For example, the ability to support protected traffic is an important requirement for many applications. This requires setting up primary and backup connections that are physically disjoint. As discussed in Chapter 2, a network with high MCLC value does not guarantee the existence of physically disjoint paths. Therefore, metrics based on maximum cross-layer disjoint paths or minimum survivable path set (defined in Section 2.2) may be more appropriate in this setting. Lightpath routings that are optimized for these metrics may potentially have different structures from the ones observed in this thesis.

Another possible future direction is to extend the current network model to a capacitated setting. Even though two different lightpath routings may tolerate the same number of physical failures from the connectivity standpoint, the impact of such failures on the capacity of the logical topology can be different. Therefore, an interesting problem is to design lightpath routing algorithms that also take the network capacity and client traffic pattern into account. For example, in [60], an ILP is developed to compute lightpath routings that allow the logical network to support a given traffic matrix under single link failures. It would be interesting to study how to extend the result in the context of multiple failures.

Finally, this thesis focuses on the design of lightpath routing that maximizes survivability, assuming the physical and logical topologies are given. Conceivably, a careful choice of the physical and logical topologies will make this lightpath routing problem easier. Therefore, the design of physical and logical topologies is an equally important problem. Conjecture 1 in Section 4.4.1, which describes a special condition for the existence of uniformly optimal lightpath routing, would be a good starting point to attack this problem area.

Bibliography

[1] Qwest Communications. https://www.qwest.com/business/resource-center/network-maps/list.html.

[2] R. G. Addie and R. Taylor. An algorithm for calculating the availability and mean time to repair for communication through a network. *Computer Networks and ISDN Systems*, 20, 1989.

[3] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, and M. Sudan. Efficient routing in optical networks. *Journal of the ACM*, 43(6), November 1996.

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

[5] J. Anderson, B.T. Doshi, S. Dravida, and P. Harshavardhana. Fast restoration of ATM networks. *Selected Areas in Communications, IEEE Journal on*, 12(1):128–138, January 1994.

[6] J. Armitage, O. Crochat, and J.-Y. Le Boudec. Design of a survivable WDM photonic network. In *IEEE INFOCOM*, April 1997.

[7] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 222–231, New York, NY, USA, 2004. ACM.

[8] Chadi M. Assi, Yinghua Ye, Abdallah A. Shami, Ibrahim Habib, Mohamed A. Ali, and Sudhir Dixit. Designing a survivable IP-over-WDM network. In *Optical Networking and Communications (OptiComm 2001)*, pages 1–11, 2001.

[9] Yontha Ath and Milton Sobel. Counterexamples to conjectures for uniformly optimally reliable graphs. *Probab. Eng. Inf. Sci.*, 14(2):173–177, 2000.

[10] Yontha Ath and Milton Sobel. Some conjectured uniformly optimal reliable networks. *Probab. Eng. Inf. Sci.*, 14(3):375–383, 2000.

[11] G. Ausiello, A. D'Atri, and M. Protasi. Structure preserving reductions among convex optimization problems. *J. Comput. System Sci.*, 21:136–153, 1980.

[12] M. Ball. Complexity of network reliability computations. *Networks*, 10:153–165, 1980.

[13] M. Ball and J. Scott Provan. Bounds on the reliability polynomial for shellable independence systems. *SIAM J. Alg. Disc. Meth.*, 3:166–181, 1982.

[14] D. Banerjee and B. Mukherjee. A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. *IEEE Journal on Selected Areas in Communications*, 14(5):903–908, June 1996.

[15] Richard A. Barry and Pierre A. Humblet. Models of blocking probability in all-optical networks with and without wavelength changers. *IEEE Journal on Selected Areas in Communications*, 14(5):858–867, 1996.

[16] D. Bauer, F. Boesch, C. Suffel, and R. Van Slyke. On the validity of a reduction of reliable network design to a graph extremal problem. *IEEE Transactions on Circuits and Systems*, 34(12):1579–1581, December 1987.

[17] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, 1997.

[18] Ramesh Bhandari. Optimal diverse routing in telecommunication fiber networks. In *IEEE INFOCOM*, June 1994.

[19] Ramesh Bhandari. *Survivable Networks: Algorithms for Diverse Routing.* Kluwer Academic Publishers, 1999.

[20] F. Boesch. On unreliability polynomials and graph connectivity in reliable network synthesis. *Journal of Graph Theory*, 10(3):339–352, 1986.

[21] F. Boesch, X. Li, and C. Suffel. On the existence of uniformly optimally reliable networks. *Networks*, 21:181–194, 1991.

[22] Béla Bollobás. *Combinatorics: set systems, hypergraphs, families of vectors, and combinatorial probability*, page 34. Cambridge University Press, New York, NY, USA, 1986.

[23] R. Braden. Requirements for internet hosts - communication layers. *RFC 1122*, 1989.

[24] T. Brecht. Lower bounds for two-terminal network reliability. Master's thesis, University of Waterloo, Waterloo, ON, 1985.

[25] Guo-Ray Cai and Yu-Geng Sun. The minimum augmentation of any graph to a k-edge-connected graph. *Networks*, 19(1):151–172, January 1989.

[26] Ruay-Shiung Chang and Leu Shing-Jiuan. The minimum labeling spanning trees. *Information Processing Letters*, 63(5):277–282, 1997.

[27] M. Chari, T. Feo, and J. Scott Provan. The delta-wye approximation procedure for two-terminal reliability. *Operations Research*, 44(5):745–757, 1996.

[28] S. Chaudhuri, G. Hjalmtysson, and J. Yates. Control of lightpaths in an optical network. In *Optical Internetworking Forum OIF2000.04*. IETF Internet Draft, January 2000. http://www.research.att.com/areas/opticalnetworking/papers/oif2000-004.doc.

[29] I Chlamtac, A Ganz, and G Karmi. Lightpath communications: a novel approach to high bandwidth optical WANs. *IEEE Transactions on Communications*, 40(7):1171–1182, 1992.

[30] Hongsik Choi, Suresh Subramaniam, and Hyeong-Ah Choi. Loopback recovery from double-link failures in optical mesh networks. *IEEE/ACM Trans. Netw.*, 12(6):1119–1130, 2004.

[31] C. Colbourn and D. Harms. Bounding all-terminal reliability in computer networks. *Networks*, 18:1–12, 1988.

[32] C. J. Colbourn. *The Combinatorics of Network Reliability.* Oxford University Press, 1987.

[33] O. Crochat and J.-Y. Le Boudec. Design protection for WDM optical networks. *IEEE Journal on Selected Areas in Communications*, 16(7), September 1998.

[34] O. Crochat, J.-Y. Le Boudec, and O. Gerstel. Protection interoperability for WDM optical networks. *IEEE/ACM Transactions on Networking*, 8(3), June 2000.

[35] P. Datta and A.K. Somani. Diverse routing for Shared Risk Resource Groups (SRRG) failures in WDM optical networks. pages 120–129, October 2004.

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

[37] John Doucette and Wayne D. Grover. Capacity design studies of span-restorable mesh transport networks with Shared-Risk Link Group (SRLG) effects. In *Optical Networking and Communications (OptiComm 2002)*, pages 25–38, 2002.

[38] G. Ellinas, A.G. Hailemariam, and T.E. Stern. Protection cycles in mesh WDM networks. *Selected Areas in Communications, IEEE Journal on*, 18(10):1924–1937, October 2000.

[39] N. Fard and Tae-Han Lee. Spanning tree approach in all-terminal network reliability expansion. *Computer Communications*, 24(13):1348–1353, 2001.

[40] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1, page 151. John Wiley & Sons, 1968.

[41] G. Fishman. A comparison of four Monte Carlo methods for estimating the probability of $s - t$ connectedness. *IEEE Trans. Reliability*, 35(2):145–155, 1986.

[42] G. Fishman. A Monte Carlo sampling plan for estimating network reliability. *Oper. Res.*, 34(4):581–594, 1986.

[43] András Frank. Connectivity augmentation of networks: structures and algorithms. *Mathematical Programming*, 84:439–441, 1999.

[44] A. Fumagalli, I. Cerutti, M. Tacca, F. Masetti, R. Jagannathan, and S. Alagar. Survivable networks based on optimal routing and WDM self-healing rings. volume 2, pages 726–733, March 1999.

[45] A. Fumagalli and L. Valcarenghi. IP restoration vs. WDM protection: is there an optimal choice? *Network, IEEE*, 14(6):34–41, November 2000.

[46] Harold N. Gabow. Two algorithms for generating weighted spanning trees in order. *SIAM Journal on Computing*, 6(1):139–150, 1977.

[47] Harold N. Gabow and Eugene W. Myers. Finding all spanning trees of directed and undirected graphs. *SIAM Journal on Computing*, 7(3):280–287, 1978.

[48] M. Garey and D. Johnson. *Computers and Intractibility, a Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., 1979.

[49] A. Gersht, S. Kheradpir, and A. Shulman. Dynamic bandwidth-allocation and path-restoration in SONET self-healing networks. *Reliability, IEEE Transactions on*, 45(2):321–331, June 1996.

[50] O. Gerstel and R. Ramaswami. Optical layer survivability: a services perspective. *Communications Magazine, IEEE*, 38(3):104–113, March 2000.

[51] O. Gerstel and R. Ramaswami. Optical layer survivability-an implementation perspective. *Selected Areas in Communications, IEEE Journal on*, 18(10):1885–1899, October 2000.

[52] O. Gerstel, G. Sasaki, S. Kutten, and R. Ramaswami. Dynamic wavelength allocation in optical networks. In *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing*, 21–24 August 1997.

[53] H. AboElFotoh and C. Colbourn. Computing 2-terminal reliability for radio-broadcast networks. *IEEE Trans. Reliability*, 38(5):538–555, December 1989.

[54] J. Hastad. Clique is hard to approximate within $n^{(1-\epsilon)}$. *Acta Math.*, 182:105–142, 1999.

[55] Tsan-Sheng Hsu, Torsten Suel, Roberto Tamassia, and Honghua Yang During The. Graph augmentation and related problems: Theory and practice, 1993.

[56] J. Q. Hu. Diverse routing in optical mesh networks. *IEEE Transactions on Communications*, 51(3), March 2003.

[57] Gianluca Iannaccone, Chen nee Chuah, Supratik Bhattacharyya, and Christophe Diot. Feasibility of IP restoration in a tier-1 backbone. *IEEE Network*, 18:13–19, 2004.

[58] R.R. Iraschko, M.H. MacGregor, and W.D. Grover. Optimal capacity placement for path restoration in STM or ATM mesh-survivable networks. *Networking, IEEE/ACM Transactions on*, 6(3):325–336, June 1998.

[59] Bill Jackson and Tibor Jordán. Connectivity augmentation of graphs. *Electronic Notes in Discrete Mathematics*, 5:185–188, 2000. 6th International Conference on Graph Theory.

[60] Daniel Dao-Jun Kan. Design of survivable IP-over-WDM networks: Providing protection and restoration at the electronic layer. *Master's Thesis, Massachusetts Institute of Technology*, June 2003.

[61] Sanjiv Kapoor and H. Ramesh. Algorithms for enumerating all spanning trees of undirected and weighted graphs. *SIAM Journal on Computing*, 24(2):247–265, 1995.

[62] D. Karger. A randomized fully polynomial time approximation scheme for the all terminal reliability problem. *SIAM Review*, 43(3):499–522, 2001.

[63] R. Karp and M. Luby. Monte Carlo algorithms for planar multiterminal network reliability problems. *Journal of Complexity*, 1(1):45–64, 1977.

[64] R. Karp and M. Luby. Monte-Carlo algorithms for enumeration and reliability problems. In *Ann. Symp. on Found. of Comp. Sci.*, November 1983.

[65] S. Kirkpatrick, Jr C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[66] M. Kurant and P. Thiran. Survivable routing in IP-over-WDM networks in the presence of multiple failures. In *EuroNGI Workshop on Traffic Engineering, Protection and Restoration for NGI*, May 2006.

[67] Maciej Kurant and Patrick Thiran. Survivable routing of mesh topologies in IP-over-WDM networks by recursive graph contraction. *IEEE Journal on Selected Areas in Communications*, 25(5), 2007.

[68] A. Kvalbein, A.F. Hansen, T. Cicic, S. Gjessing, and O. Lysne. Fast recovery from link failures using resilient routing layers. pages 554–560, June 2005.

[69] Ching Law and Kai-Yeung Siu. Online routing and wavelength assignment in single-hub WDM rings. *IEEE Journal on Selected Areas in Communications*, 18(10), October 2000.

[70] Kayi Lee and Eytan Modiano. Cross-layer survivability in WDM-based networks. In *IEEE INFOCOM*, Rio de Janeiro, Brazil, April 2009.

[71] S. Lee, Yinzhe Yu, S. Nelakuditi, Zhi-Li Zhang, and Chen-Nee Chuah. Proactive vs reactive approaches to failure resilient routing. volume 1, page 186, March 2004.

[72] Guangzhi Li, Robert Doverspike, and Charles Kalmanek. Fiber span failure protection in mesh optical networks. In *Optical Networking and Communications (OptiComm 2001)*, pages 130–141, 2001.

[73] G. Lingner, T. Politof, and A. Satyanarayana. A forbidden minor characterization and reliability of a class of partial 4-trees. *Networks*, 25:139–146, 1995.

[74] László Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematcis*, 13:383–390, 1975.

[75] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994.

[76] Eytan Modiano and Aradhana Narula-Tam. Survivable lightpath routing: A new approach to the design of WDM-based networks. *IEEE Journal on Selected Areas in Communications*, 20(4), May 2002.

[77] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[78] K. Murakami and H. Kim. Virtual path routing for survivable ATM networks. *IEEE/ACM Transactions on Networking*, 4(1), February 1996.

[79] W. Myrvold, K. Cheung, L. Page, and J. Perry. Uniformly-most reliable networks do not always exist. *Networks*, 21:417–419, 1991.

[80] Wendy Myrvold. Reliable network synthesis: Some recent developments. In *International Conference on Graph Theory, Combinatorics, Algorithms, and Applications, volume II*, pages 650–660, 1996.

[81] Aradhana Narula-Tam, Eytan Modiano, and Andrew Brzezinski. Physical topology design for survivable routing of logical rings in WDM-based networks. *IEEE Journal on Selected Areas in Communications*, October 2004.

[82] L. Nel and C. Colbourn. Combining Monte Carlo estimates and bounds for network reliability. *Networks*, 20:277–298, 1990.

[83] E. Neufeld and C. Colbourn. The most reliable series-parallel networks. *Networks*, 15:27–32, 1985.

[84] E. Oki, N. Matsuura, K. Shiomoto, and N. Yamanaka. A disjoint path selection scheme with Shared Risk Link Groups in GMPLS networks. *IEEE Communications Letters*, 6(9):406–408, September 2002.

[85] A. E. Ozdaglar and D. P. Bertsekas. Routing and wavelength assignment in optical networks. *IEEE/ACM Transactions on Networking*, 11(2):259–272, April 2003.

[86] T. Politof, A. Satyanarayana, and L. Tung. An $O(n \cdot \log(n))$ algorithm to compute the all-terminal reliability of $(K_5, K_{2,2,2})$ free networks. *IEEE Trans. Reliability*, 41(4):512–517, December 1992.

[87] J. Scott Provan and M. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM J. Comput.*, 12(4):777–788, November 1983.

[88] S. Finn R. Barry and M. Medard. WDM loop-back in mesh networks. In *IEEE INFOCOM*, 1999.

[89] R. Van Sylke and H. Frank. Network reliability analysis: Part I. *Networks*, 1:279–290, 1972.

[90] Prabhakar Raghavan and Clark D. Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

[91] Bala Rajagopalan, Dimitrios Pendarakis, Debanjan Saha, Ramu Ramamoorthy, and Krishna Bala. IP over optical networks: Architectural aspects. *IEEE Communications Magazine*, September 2000.

[92] S. Ramamurthy and B. Mukherjee. Survivable WDM mesh networks, part I - protection. In *IEEE INFOCOM*, 1999.

[93] S. Ramamurthy and B. Mukherjee. Survivable WDM mesh networks, part II - restoration. In *International Conference on Communications*, 1999.

[94] A. Ramanathan and C. Colbourn. Counting almost minimum cutsets with reliability applications. *Mathematical Programming*, 39:253–261, 1987.

[95] R. Ramaswami and K. N. Sivarajan. Routing and wavelength assignment in all-optical networks. *IEEE/ACM Transactions on Networking*, 3(5), October 1995.

[96] R. C. Read and R. E. Tarjan. Bounds on backtrack algorithms for listing cycles, paths, and spanning trees. *Networks*, 6(1):139–150, 1975.

[97] Dirk P. Kroese Reuven Y. Rubinstein. *Simulation and the Monte Carlo method.* John Wiley and Sons Inc., 2007.

[98] A. Rosenthal and D. Frisque. Transformations for simplifying network reliability calculations. *Networks*, 7:97–111, 1977.

[99] L. Sahasrabuddhe, S. Ramamurthy, and B. Mukherjee. Fault management in IP-over-WDM networks: WDM protection versus IP restoration. *Selected Areas in Communications, IEEE Journal on*, 20(1):21–33, January 2002.

[100] P. Sebos, J. Yates, G. Hjalmtysson, and A. Greenberg. Auto-discovery of Shared Risk Link Groups. In *Optical Fiber Communication Conference*, March 2001.

[101] Arunabha Sen, Bin Hao, and Bao Hong Shen. Survivable routing in WDM networks. In *Proceedings of the Seventh International Symposium on Computers and Communications*, page 726, Washington, DC, USA, 2002. IEEE Computer Society.

[102] Arunabha Sen, Bin Hao, and Bao Hong Shen. Minimum cost ring survivability in WDM networks. In *Workshop on High Performance Switching and Routing*, pages 183–188, 2003.

[103] Salman Z. Shaikh. Span-disjoint paths for physical diversity in networks. In *Proc. IEEE Symposium on Computers and Communications*, 1995.

[104] Gangxiang Shen and W.D. Grover. Extending the p-cycle concept to path segment protection for span and node failure recovery. *Selected Areas in Communications, IEEE Journal on*, 21(8):1306–1319, October 2003.

[105] L. Shen, X. Yang, and B. Ramamurthy. Shared Risk Link Group (SRLG)-diverse path provisioning under hybrid service level agreements in wavelength-routed optical mesh networks. *Networking, IEEE/ACM Transactions on*, 13(4):918–931, August 2005.

[106] A. Shooman and A. Kershenbaum. Exact graph-reduction algorithms for network reliability analysis. In *IEEE GLOBECOM*, December 1991.

[107] A. Shooman and A. Kershenbaum. Methods for communication-network reliability analysis: Probabilistic graph reduction. In *Annual Reliability and Maintainability Symposium*, 1992.

[108] Martin L. Shooman. *Reliability of computer systems and networks: fault tolerance, analysis and design*, pages 309–318. John Wiley and Sons, 2002.

[109] D. Stamatelakis and W.D. Grover. IP layer restoration and network planning based on virtual protection cycles. *Selected Areas in Communications, IEEE Journal on*, 18(10):1938–1949, October 2000.

[110] Suresh Subramaniam, Murat Azizoglu, and Arun K. Somani. All-optical networks with sparse wavelength conversion. *IEEE/ACM Trans. Netw.*, 4(4):544–557, 1996.

[111] C. Tanguy. Exact two-terminal reliability of directed networks. In *DRCN*, La Rochelle, France, October 2007.

[112] K. Thulasiraman, M.S. Javed, and Guoliang Xue. Circuits/cutsets duality and a unified algorithmic framework for survivable logical topology design in IP-over-WDM optical networks. pages 1026–1034, April 2009.

[113] Ajay Todimala and Byrav Ramamurthy. Survivable virtual topology routing under Shared Risk Link Groups in WDM networks. In *BROADNETS '04: the First International Conference on Broadband Networks*, pages 130–139, Washington, DC, USA, 2004. IEEE Computer Society.

[114] L. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, August 1979.

[115] P.-J. Wan, L.-W. Liu, and O. Frieder. Grooming of arbitrary traffic in SONET/WDM rings. In *IEEE GLOBECOM*, December 1999.

[116] G. Wang. A proof of Boesch's conjecture. *Networks*, 24:277–284, 1994.

[117] O.J. Wasem. An algorithm for designing rings for survivable fiber networks. *Reliability, IEEE Transactions on*, 40(4):428–432, October 1991.

[118] Ondria J. Wasem. Optimal topologies for survivable fiber optics networks using SONET self-healing rings. In *IEEE GLOBECOM*, 1991.

[119] T. Watanabe, Y. Higashi, and A. Nakamura. An approach to robust network construction from graph augmentation problems. volume 4, pages 2861–2864, May 1990.

[120] Gaoxi Xiao and Xiaoshan Pan. Heuristic for the maximum disjoint paths problem in wavelength-routed networks with Shared-Risk Link Groups. *Journal of Optical Networking*, 3(1), January 2004.

[121] Dahai Xu, Yizhi Xiong, and Chunming Qiao. A new PROMISE algorithm in networks with Shared Risk Link Groups. In *IEEE GLOBECOM*, December 2003.

[122] Dahai Xu, Yizhi Xiong, Chunming Qiao, and Guangzhi Li. Trap avoidance and protection schemes in networks with Shared Risk Link Groups. *IEEE Journal of Lightwave Technology*, 21(11), November 2003.

[123] Jin Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.

[124] Shengli Yuan, S. Varma, and Jason P. Jue. Minimum-color path problems for reliability in mesh networks. In *INFOCOM*, pages 2658–2669, 2005.

[125] Hui Zang, C. Ou, and B. Mukherjee. Path-protection routing and wavelength assignment in WDM mesh networks under duct-layer constraints. *IEEE/ACM Transactions on Networking*, 11, April 2003.

[126] X. Zhang and C. Qiao. An effective and comprehensive approach to traffic grooming and wavelength assignment in SONET/WDM rings. *IEEE/ACM Transactions on Networking*, 8(5), October 2000.

[127] H. Zimmermann. OSI reference model—the ISO model of architecture for open systems interconnection. pages 2–9, 1988.