Reliability in Layered Networks with Random Link Failures

Kayi Lee, Hyang-Won Lee and Eytan Modiano
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: {kylee, hwlee, modiano}@mit.edu

***Abstract*—We consider network reliability in layered networks where the lower layer experiences random link failures. In layered networks, each failure at the lower layer may lead to multiple failures at the upper layer. We generalize the classical polynomial expression for network reliability to the multi-layer setting. Using random sampling techniques, we develop polynomial time approximation algorithms for the failure polynomial. Our approach gives an approximate expression for reliability as a function of the link failure probability, eliminating the need to resample for different values of the failure probability. Furthermore, it gives insight on how the routings of the logical topology on the physical topology impact network reliability. We show that maximizing the min cut of the (layered) network maximizes reliability in the low failure probability regime. Based on this observation, we develop algorithms for routing the logical topology to maximize reliability.**

I. Introduction

Modern communication networks are constructed using a layered approach, with one or more electronic layers (e.g., IP, ATM, SONET) built on top of an optical fiber network. The survivability of such networks under fiber failures largely depends on how the logical electronic topology is embedded onto the physical fiber topology; called *lightpath routing*. However, assessing reliability performance achieved by a lightpath routing can be rather challenging because seemingly independent logical links can share the same physical link, which can lead to correlated failures. In this paper, we focus on assessing the reliability of multi-layer networks assuming that physical links experience random failures.

A natural survivability metric in this context is the probability that given a lightpath routing, the logical topology remains connected; we call this probability the *cross-layer (network) reliability*. The cross-layer reliability reflects the survivability performance achieved by the lightpath routing. Unfortunately, computing reliability in a single-layer setting is known to be $\#P$-complete [1]. Moreover, it is also $\#P$-complete even to approximate the reliability up to ϵ relative accuracy [2]. Therefore, computing cross-layer reliability is likely to be a very difficult problem (it is at least as difficult as the single-layer problem, and hence, also $\#P$-complete). Although there has been a large body of work on estimating single-layer network reliability (defined as the probability that a single-layer network remains connected under random link failures), there is no known work dealing with cross-layer reliability.

This motivates us to study the cross-layer reliability problem. In particular, we first extend the polynomial expression for single-layer network reliability to the layered setting. This polynomial (cf. (1)) 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, we develop a Monte Carlo simulation based estimation method 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.

Another interesting property of the polynomial expression 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 particular, we design lightpath routing algorithms with the goal of increased reliability and evaluate their performance using the reliability estimation techniques developed in this paper. The idea behind these algorithms is to maximize the minimum number of failures that lead to the disconnection of the (layered) network. We observe that this approach can achieve good reliability performance in the low failure probability regime. We provide extensive simulation results to verify our algorithms and insights.

II. Previous Work

The network reliability estimation problem has been extensively studied in the single-layer setting. Since it is a difficult problem, 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 [3]–[10], the applications

This work was supported by NSF grants CNS-0626781 and CNS-0830961 and by DTRA grants HDTRA1-07-1-0004 and HDTRA-09-1-005.

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 [11]–[13]. Using simulation, the reliability can be approximated to an arbitrary accuracy, but the number of required iterations tends to be very large especially when the failure probability is small. Moreover, the simulation must be repeated for different values of the failure probability.

Another approach is to use a polynomial expression for reliability [15] 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 [16]–[20]. This approach is computationally attractive, but its estimation accuracy is not guaranteed. [14] estimated the reliability by counting the small cut sets. In [21], a random sampling technique is used to enhance those bounding results. In particular, [21] considers another form of the polynomial used in [22], and estimates some of the coefficients by enumerating spanning trees in the graph. These estimates are used to improve the algebraic bound in [22]. This approach is relevant to our work in that it tries to approximate the coefficients in the polynomial through random sampling. However, the tree-sampling techniques of [21] are not applicable to our multi-layer setting because spanning-trees in the multi-layer setting exhibit fundamentally different characteristics from the single-layer counterparts [23].

Our work is the first attempt to address reliability estimation in multi-layer networks. Although some of the methods in a single-layer setting may be applicable to the layered setting, they show limited performance in terms of the number of required iterations or the accuracy of estimation. The most relevant work in multi-layer setting is our work in [24] where a new survivability metric called the Min Cross Layer Cut (MCLC) is introduced by generalizing the traditional min cut. The MCLC is defined as the minimum number of physical cuts that are needed to disconnect the logical topology, and it is closely related to cross-layer reliability. As MCLC grows, it becomes harder to disconnect the logical topology, which in turn implies high reliability. Consequently, a lightpath routing that maximizes the MCLC may achieve good reliability performance. In fact, we will use some of the lightpath routing algorithms from [24] to demonstrate our reliability estimation algorithm. Thus, our work in this paper can be viewed as an extension of [24] to the case of random link failures.

III. Model and Background

We consider a layered network $\mathcal{G}$ that 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, where V and E are the set of nodes and links respectively. In the context of WDM network, a logical link is called a *lightpath*, and each lightpath is routed over the physical topology. This *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.

Each physical link fails independently with probability p, and 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 set S of physical links is called a *cross-layer cut* if and only if the failure of the links in S causes the logical network to be disconnected. We also define the *network state* as the subset of physical links S that failed. Hence, if S is a cross-layer cut, the network state S represents a *disconnected* network state. Otherwise, it is a *connected* state.

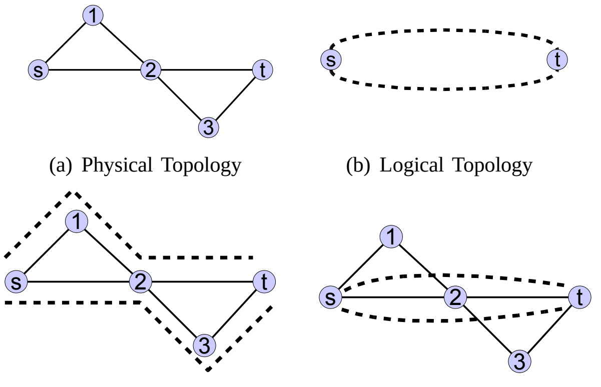
A. Importance of Lightpath Routing

The reliability of a multi-layer network is defined as the probability that the logical network remains connected. It is important to note that the reliability depends on the underlying lightpath routing. For example, in Figure 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 routes the two logical links using 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, which routes the two logical links over the same 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 depends on the value of p.

B. Cross-Layer Failure Polynomial

In general, it is difficult to compare lightpath routings since computing the cross-layer reliability is $\#P$-Complete. Hence, we first need to develop an algorithm that can accurately estimate the reliability. However, even if an estimation algorithm is capable of comparing lightpath routings for a certain value of p, it does not mean that the comparison is valid at other values (as shown in the above example), and this may require rerunning the algorithm when the probability of interest p has changed. Hence, 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. Having a failure polynomial makes it possible to compare reliability among different lightpath routings at all values of p. Furthermore, as discussed below, the failure polynomial gives insights to the design of lightpath routings for better reliability.

We introduce a polynomial expression for reliability which is a natural extension of the single-layer polynomial [15] to the cross-layer setting. Assume that there are m physical links, i.e., $|E_P| = m$. The probability associated with a network



(a) Physical Topology (b) Logical Topology

(c) Disjoint Routing (d) Non-Disjoint Routing

Fig. 1. Non-disjoint routings can sometimes be more reliable

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 disconnected network states S with $|S| = i$, then the probability that the network gets disconnected is simply the sum of the probabilities over all disconnected states, i.e.,

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

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) [24] 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 [24], 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 paper, 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 VIII, where we present some lightpath routing algorithms for achieving maximum reliability. A similar observation was made for single-layer networks in [25].

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

Definition 1 (Relative Approximation): A function $\hat{F}(p)$ is an ϵ-approximation for the failure polynomial $F(p)$ if

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

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 [2]. 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 2 ((ϵ, δ)-approximation): A function $\hat{F}(p)$ is an (ϵ,δ)-approximation for the failure polynomial $F(p)$ if

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

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

C. 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., disconnected network states). 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, and counts the number of times N such that $e \in G$. The Monte Carlo method returns the quantity $\frac{N}{T}|U|$ as an estimator of $|G|$. The Estimator Theorem states that:

*Theorem 1 (***Estimator Theorem*** [26]):* Let $\rho = \frac{|G|}{|U|}$. Then the Monte Carlo method yields an ϵ-approximation to $|G|$ with probability at least 1-δ, provided that

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

In other words, if we sample from the ground set U frequently enough, we can estimate $|G|$ accurately with high probability. According to Theorem 1, the ratio ρ, called the *density* of the set G, is inversely proportional to the required number of iterations T. 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.

IV. Estimating Cross-Layer Reliability

Our approach to approximating the cross-layer failure polynomial is to estimate the values of N_i in Equation (1). 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 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 $0 \le p \le 1$,

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

■

Corollary 1: 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 2, A gives an (ϵ, δ)-approximation algorithm for the failure polynomial. ■

Therefore, for the rest of the section, our goal is to estimate each N_i with ϵ accuracy and probability at least $1 - \frac{\delta}{m+1}$.

A. 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:

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

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 (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.

B. Lower Bounding ρ_i

Given a layered network, suppose its Min Cross Layer Cut value d is known, Theorem 3 gives a lower bound on ρ_i:

Theorem 3: For $i \ge d, \rho_i \ge \frac{\binom{m-d}{i-d}}{\binom{m}{i}}$.

Proof: Since d is the Min Cross Layer Cut value, there exists a cross-layer cut S of 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 $N_i \ge \binom{m-d}{i-d}$, 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 (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 V, we will present an enhanced version 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.

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

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

where the terms inside the $\min$ operator are the number of iterations required by enumeration and Monte Carlo methods respectively. Overall, only $O(m^{d+1} \log m)$ iterations are required. The improvement in running time of this combined approach is illustrated by Figure 2.

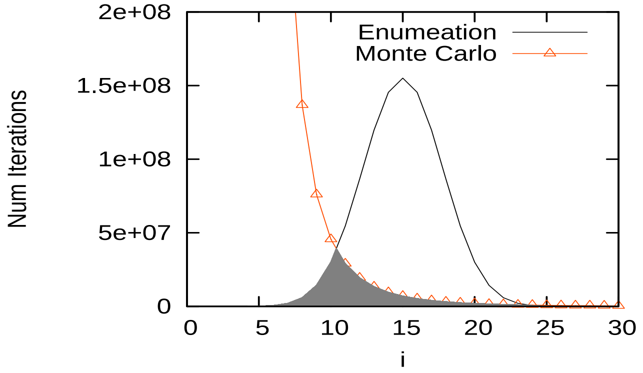


Fig. 2. Monte-Carlo vs Enumeration: Number of iterations for estimating N_i, for a network with 30 physical links, $\epsilon = 0.01, \delta = \frac{0.01}{31}, d = 4$. The shaded region represents the required iterations for the combined approach.

V. An Enhanced Reliability Estimation Algorithm

In the previous section, we presented an algorithm that combines the enumeration and Monte Carlo methods to estimate the values of N_i. For the Monte Carlo method, the number of iterations required is inversely proportional to $\tilde{\rho}_i$, the lower bound for ρ_i given by Theorem 3. In this section, we introduce a probabilistic method to derive a much tighter lower bound for ρ_i, which significantly reduces the number of iterations required to estimate N_i, and therefore speeds up the overall running time of the algorithm.

Let d be the Min Cross Layer Cut value of the network. For any $i > d$, suppose $\mathcal{C}^i = \{C_1, \ldots, C_n\}$ is a collection of cross-layer cuts, where each cross-layer cut C_j has size less than i. For each $C_j \in \mathcal{C}^i$, let $\mathcal{L}_j^i$ be the family of supersets of C_j with i physical links. Finally, let $\mathcal{L}^i = \cup_{C_j \in \mathcal{C}^i} \mathcal{L}_j^i$ be the union over all $\mathcal{L}_j^i$. The enhanced reliability estimation algorithm is based on the following key observation:

Theorem 4: $\rho_i \geq \frac{|\mathcal{L}^i|}{\binom{m}{i}}$.

Proof: Every set $S \in \mathcal{L}_j^i$ is a superset of the cross-layer cut C_j, and is therefore a cross-layer cut of size i. This implies every set S in $\mathcal{L}^i$ is also a cross-layer cut with size i. Therefore, we have $|\mathcal{L}^i| \leq N_i$. It follows that $\frac{|\mathcal{L}^i|}{\binom{m}{i}} \leq \frac{N_i}{\binom{m}{i}} = \rho_i$. ■

Therefore, if we know the value of $|\mathcal{L}^i|$, we can use $\frac{|\mathcal{L}^i|}{\binom{m}{i}}$ as the lower bound for ρ_i in the Monte Carlo method to estimate N_i. Note that if $\mathcal{C}^i$ contains only a Min Cross Layer Cut of the network, the value of $\frac{|\mathcal{L}^i|}{\binom{m}{i}}$ is equal to the bound given by Theorem 3. In addition, the size of the union $\mathcal{L}^i$ is monotonic in $\mathcal{C}^i$. Therefore, we can improve the lower bound by adding more cross-layer cuts to $\mathcal{C}^i$.

To apply this result to reliability estimation, we can modify our algorithm presented in the previous section to also maintain the collection $\mathcal{C}$ of cross-layer cuts as we carry out the enumeration or Monte Carlo methods. Specifically, as we discover a cross-layer cut C_j of size i when estimating N_i, we will add the cut C_j to our collection $\mathcal{C}$. When we move on to estimate N_{i+1}, we will have a collection $\mathcal{C}$ of cross-layer cuts with size i or smaller. We can therefore apply Theorem 4 to obtain a better lower bound for estimating N_{i+1}.

A. Computing $|\mathcal{L}^i|$

Although the value of each $|\mathcal{L}_j^i| = \binom{m-|C_j|}{i-|C_j|}$ can be computed easily, finding the size of the union $\mathcal{L}^i = \cup_{C_j \in \mathcal{C}^i} \mathcal{L}_j^i$ can be difficult because L_j^i's are not disjoint. This problem can be formulated as the Union of Sets Problem [27], where Monte Carlo methods exist to estimate the size of $|\mathcal{L}^i|$. We summarize the result as follows. Please refer to [23] for details.

Theorem 5: Let $\mathcal{C} = \{C_1, \ldots, C_n\}$ be a collection of cross-layer cuts of the layered network. For each $C_j \in \mathcal{C}$, let $\mathcal{L}_j^i$ be the family of supersets of C_j with i physical links. There exists a Monte Carlo method that produces an $(\epsilon_{lb}, \delta_{lb})$-approximation, $\hat{L}^i$, for $|\cup_{C_j \in \mathcal{C}} \mathcal{L}_j^i|$, provided that the number of samples is at least:

$$T_{lb} = \frac{4|\mathcal{C}|}{\epsilon_{lb}^2} \ln \frac{2}{\delta_{lb}}. \tag{3}$$

Proof: Let $U = \left\{(S, j) : S \in \mathcal{L}^i, j \in \{1, \ldots, |\mathcal{C}|\}, C_j \subset S\right\}$ be the ground set of the Monte Carlo algorithm, and let $G = \left\{(S, j) : S \in \mathcal{L}^i, j = \min\{k : C_k \subset S\}\right\}$ be the events of interest. We show in [23] that the ground set U can be sampled uniformly at random. Since $|G| = |\mathcal{L}^i|$ and $\frac{|G|}{|U|} \geq \frac{1}{|\mathcal{C}|}$, Theorem 5 follows immediately from the Estimator Theorem.

■

Theorem 5 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 6: Let $\hat{L}^i$ be an $(\epsilon_{lb}, \delta_{lb})$-approximation for $|\cup_{j=1}^m \mathcal{L}_j^i|$. Then, the Monte Carlo method described in Section IV-A yields an $(\epsilon_{mc}, \delta_{lb} + \delta_{mc})$-approximation for N_i, provided that the number of samples is at least:

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

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 for 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. ■

We would like to conclude this section by discussing the tradeoff between the running times T_{lb} and T_{mc} in Equations (3) and (4). In general, the method presented in this section produces a tighter lower bound $\hat{\rho}_i$ for ρ_i with smaller ϵ_{lb} and δ_{lb} values. A tighter lower bound on ρ_i will require a smaller number of iterations T_{mc} for estimating N_i. On the other hand, as implied by Equation (3), smaller values of ϵ_{lb} and δ_{lb} require more iterations to obtain the lower bound $\hat{\rho}_i$. Therefore, there is a tradeoff in terms of running time between the lower bound estimation and the N_i estimation.

VI. 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 IV and V, 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.

We used the augmented NSFNET (Figure 3) as the physical topology. We generated 350 random logical topologies with 6 to 12 nodes and created lightpath routings using the

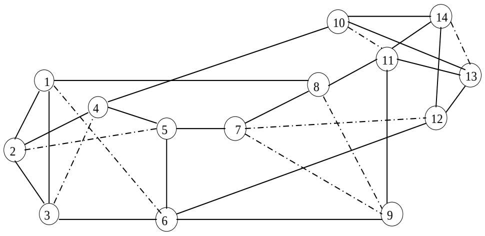


Fig. 3. The Augmented NSFNET.

MCF (Multi-Commodity Flow) algorithms described in Section VIII. For each lightpath routing, we ran three 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 IV-C, with $\epsilon = \delta = 0.01$.
3) MIXED$_{\text{enhanced}}$: The enhanced version of the algorithm introduced in Section V. 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 I 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 enhanced algorithm is able to further reduce the number of iterations. This is because it exploits the knowledge about the discovered cross-layer cuts $\mathcal{C}$ to obtain a better lower bound for each ρ_i, as shown in Figure 4.

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$_{\text{enhanced}}$	11,968,535	2,485,477	14,454,012

TABLE I
NUMBER OF ITERATIONS FOR EACH ALGORITHM

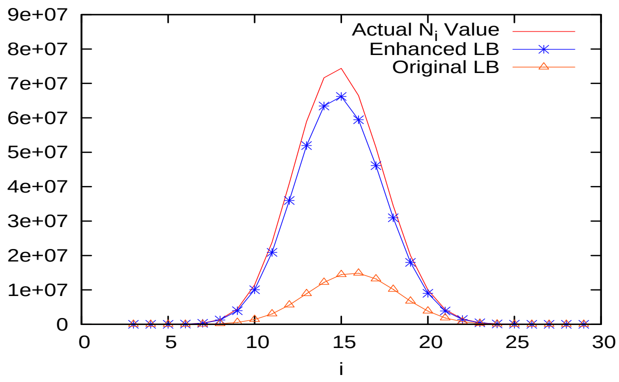


Fig. 4. Lower bounds for N_i produced by MIXED$_{\text{original}}$ and MIXED$_{\text{enhanced}}$

As a result, using Monte Carlo sampling to estimate both the lower bound for ρ_i and the value of N_i appears to be very effective in reducing the running time of the algorithm.

In our simulation running on an Intel(R) Xeon 2.50GHz CPU machine, the algorithm MIXED$_{\text{enhanced}}$ computed the failure polynomial within 30 minutes.

Finally, we compare the actual accuracy of the failure polynomial generated by algorithm MIXED$_{\text{enhanced}}$ with the theoretical guarantee given by the Estimator Theorem. Figure 5 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.

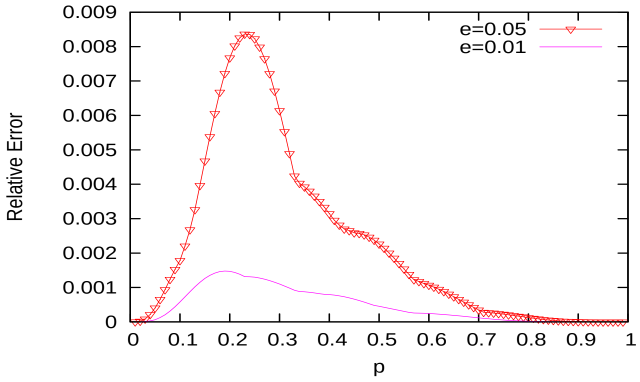


Fig. 5. Relative error of the failure polynomial approximation

VII. 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 $O(m^{d+1} \log m)$ 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:

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

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.

VIII. Lightpath Routing Algorithms for Reliability Maximization

As illustrated in Section III-A, 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 1, a lightpath routing that is optimal for a certain value of p may not perform as well for different values of p. This makes the network design aspect of cross-layer reliability a challenging and interesting problem.

We present three lightpath routing algorithms for reliability maximization. As discussed in Section III-B, maximizing the Min Cross Layer Cut (MCLC) is closely related to maximizing reliability, especially for small values of p. We will prove this claim more formally. Assume that logical and physical topologies are given. Let d^{Π} be the MCLC achieved by a lightpath routing Π, and $F^{\Pi}(p)$ be its failure polynomial.

Theorem 7: Let Π and Ψ be two lightpath routings such that $d^{\Pi} > d^{\Psi}$. For $p \leq \frac{1}{2+\binom{m}{\lfloor m/2 \rfloor}}$, $F^{\Pi}(p) \leq F^{\Psi}(p)$.

Proof: Let N_i^{Π}'s be the coefficients in the failure polynomial of Π. Then $N_i^{\Pi} \leq \binom{m}{\lfloor m/2 \rfloor}$. It follows that:

$$F^{\Pi}(p) \leq \binom{m}{\lfloor m/2 \rfloor} p^{d^{\Psi}} (1-p)^{m-d^{\Psi}} \sum_{i=d^{\Pi}-d^{\Psi}}^{m-d^{\Psi}} (\frac{p}{1-p})^i$$
$$\leq \binom{m}{\lfloor m/2 \rfloor} p^{d^{\Psi}} (1-p)^{m-d^{\Psi}} \frac{p}{1-2p}$$
$$\leq p^{d^{\Psi}} (1-p)^{m-d^{\Psi}}$$
$$\leq \sum_{i=d^{\Psi}}^{m} N_i^{\Psi} p^i (1-p)^{m-i} = F^{\Psi}(p),$$

where the first and third inequalities follow from the fact $N_i^{\Pi} \leq \binom{m}{\lfloor m/2 \rfloor}$ for all i and the assumption $p \leq \frac{1}{2+\binom{m}{\lfloor m/2 \rfloor}}$. The second inequality is due to geometric series and $d^{\Pi} - d^{\Psi} \geq 1$, and the last inequality is due to $N_i^{\Psi} \geq 1$. ■

This theorem shows that for small p, a larger MCLC value for a lightpath routing implies a higher cross-layer reliability. Consequently, maximizing MCLC is a good strategy for reliability maximization. Motivated by this observation, we present some lightpath routing algorithms that try to maximize reliability by focusing on MCLC.

In fact, in [24], we looked at the survivable lightpath routing problem with the objective of maximizing the MCLC of a layered network. The main algorithm, called multi-commodity flow based lightpath routing $\mathsf{MCF}_{\mathsf{MinCut}}$, is shown to outperform the common Shortest Path algorithm in terms of the MCLC performance. It is therefore interesting to see whether a similar observation holds in terms of reliability to random failures. We also enhance the algorithm $\mathsf{MCF}_{\mathsf{MinCut}}$ to better account for cross-layer reliability. This enhanced algorithm, called $\mathsf{MCF}_{\mathsf{MinPerCut}}$, tries to capture the connectivity structure of the layered network in greater details. Below is the brief description of the three lightpath routing algorithms we will consider.

1) Multi-Commodity Flow ($\mathsf{MCF}_{\mathsf{MinCut}}$): The multi-commodity flow lightpath routing algorithm $\mathsf{MCF}_{\mathsf{MinCut}}$ can be formulated as an integer linear program (ILP):

$$\mathsf{MCF}_{\mathsf{MinCut}}: \quad \text{Minimize } \rho, \quad \text{subject to:}$$
$$\rho \geq \sum_{(s,t) \in E_L} w(s,t) f_{ij}^{st} \qquad \forall (i,j) \in E_P$$
$$f_{ij}^{st} \in \{0,1\}$$
$$\{(i,j) : f_{ij}^{st} = 1\} \text{ forms an } (s,t)\text{-path in } G_P, \ \forall (s,t) \in E_L,$$

where $w(s,t)$ is the weight assigned to logical link (s,t). The optimal lightpath routing under this algorithm is determined by the weights $w(s,t)$. For example, if $w(s,t)$ is set to 1 for all logical links, the above formulation will minimize the number of logical links that traverse the same fiber. In other words, this uniform weight function treats each logical link equally, and seeks to minimize the impact of a single physical link failure on the number of disconnected logical links. However, the connectivity is not well captured under this function since the logical network may remain connected even when a large number of logical links fail. In order to better account for the connectivity, we use the weight function $w(s,t) = \frac{1}{MinCut_L(s,t)}$, where $MinCut_L(s,t)$ is the size of the min-cut between nodes s and t in the logical topology. Intuitively, this weight function attempts to minimize the *impact* of a single fiber failure to the logical connectivity, where impact is defined to be the total sum of weight of the logical links that traverse the fiber. Since the weight is defined to be $\frac{1}{MinCut_L(s,t)}$, a logical link that belongs to a small cut will contribute more weight than a logical link in a large cut. Presumably, if a lightpath routing ensures that any single fiber failure has a small impact to the logical connectivity, a better

MCLC will be achieved. In [24], it was shown that using the weight function $w(s,t) = \frac{1}{MinCut_L(s,t)}$ achieves better MCLC than using the uniform weight function $w(s,t) = 1$.

2) Enhanced Multi-Commodity Flow ($\mathsf{MCF}_{\mathsf{MinPerCut}}$*):* The enhanced version of the Multi-Commodity flow algorithm $\mathsf{MCF}_{\mathsf{MinPerCut}}$ tries to capture the impact of a single fiber failure in much greater details. For each fiber failure, the algorithm measures its impact on each logical cut. As a result, the algorithm tries to minimize the maximum impact of a fiber failure among all possible logical cuts. It can be formulated by a similar ILP as follows:

$$\mathsf{MCF}_{\mathsf{MinPerCut}}: \quad \text{Minimize } \rho, \quad \text{subject to:}$$

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

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

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

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 weight function $w(s,t) = \frac{1}{MinCut_L(s,t)}$ is also used here. Note that for each physical link (i,j), this formulation measures the total weight of the logical links in every logical cut $\delta(S)$ that traverse (i,j). The large number of constraints allows the formulation to capture the impact of a fiber failure in much greater details. In order to circumvent the issue of solving an integer program with a large number of constraints, we instead obtain an approximate solution by solving the relaxed integer program and using randomized rounding to obtain a lightpath routing. Please refer to [24] for more details about using randomized rounding for lightpath routing.

3) Shortest Path (SP*):* The Shortest Path algorithm independently routes each logical link (s,t) on a physical path with the minimum number of hops. Since we are assuming that every physical link fails with the same probability p, the failure probability of (s,t)-path is minimized when it is routed over the shortest path. Hence, under the algorithm SP, each lightpath greedily takes the most reliable route.

IX. Simulation Results

We used the augmented NSFNET (Figure 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 different lightpath routing algorithms mentioned in the previous section. For each lightpath routing generated by the algorithms, we compute an approximate failure polynomial using the technique proposed in Section V, and evaluate its reliability.

Figure 6 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 were able to generate more lightpath routings with high reliability than the Shortest Path Algorithm. For small p, the term $N_d p^d (1-p)^{m-d}$, where d is the Min Cross Layer Cut, dominates the failure polynomial. Therefore, maximizing d has the effect of maximizing the reliability of the network.

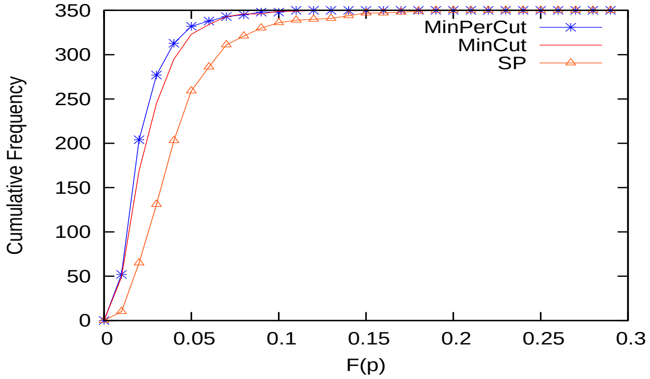


Fig. 6. Reliability CDF for different algorithms: $p = 0.1$

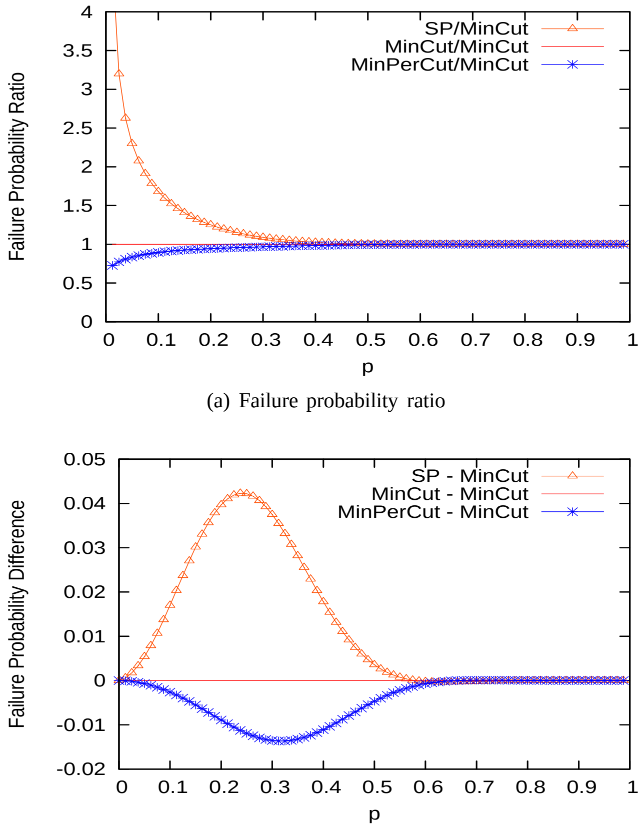
The dependence of reliability on lightpath routing and link failure probability p is further illustrated by Figure 7, which plots the ratio and absolute difference of average failure probabilities of the lightpath routings generated by the three algorithms, using $\mathsf{MCF}_{\mathsf{MinCut}}$ as the baseline. When p is small, the maximum MCLC routings are clearly better than the shortest path routing in terms of the ratio of failure probability. However, as p gets larger, the difference in reliability performance among the algorithms diminishes. In fact, as seen in Figure 7(b), the failure probabilities of all three algorithms are very close. This is because for large p, the failure probability for any lightpath routing would be very close to 1.

Figure 8 compares the average N_i values of the lightpath routings generated by the algorithms. Again using $\mathsf{MCF}_{\mathsf{MinCut}}$ as the baseline, Figure 8 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 Shortest Path Algorithm, on the other hand, has the simple objective that minimizes the total number of physical hops. In an environment where p is high, minimizing the physical hops may be a better strategy, as we have seen in Figure 1.

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).

X. Conclusion

We considered 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 could not be used for our problem. To that end, we developed a Monte Carlo simulation based estimation algorithm that approximates cross-layer reliability with high probability. We first extended 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



(a) Failure probability ratio

(b) Failure probability difference

Fig. 7. Ratio and absolute difference of average failure probabilities among different algorithms

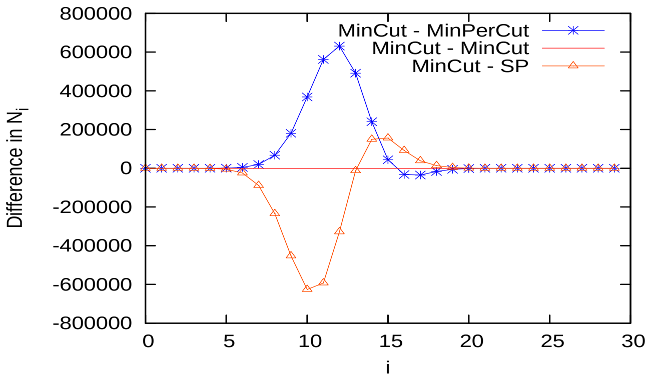


Fig. 8. Difference in average N_i values among different algorithms.

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 observed through the polynomial expression that lightpath routings that maximize the MCLC can perform very well in terms of reliability. This observation led 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 could be used to enhance the current lightpath routing. Therefore, future directions include the use our estimation algorithm for improving the lightpath routing, and the development of a new lightpath routing algorithm that maximizes the cross-layer reliability.

References

[1] L. Valiant, “The complexity of enumeration and reliability problems,” *SIAM J. Comput.*, vol. 8, no. 3, pp. 410–421, Aug. 1979.

[2] J. S. Provan and M. Ball, “The complexity of counting cuts and of computing the probability that a graph is connected,” *SIAM J. Comput.*, vol. 12, no. 4, pp. 777–788, Nov. 1983.

[3] C. Tanguy, “Exact two-terminal reliability of directed networks,” in *DRCN*, La Rochelle, France, Oct. 2007.

[4] A. Shooman and A. Kershenbaum, “Exact graph-reduction algorithms for network reliability analysis,” in *IEEE GLOBECOM*, Dec. 1991.

[5] A. Rosenthal and D. Frisque, “Transformations for simplifying network reliability calculations,” *Networks*, vol. 7, pp. 97–111, 1977.

[6] A. Shooman and A. Kershenbaum, “Methods for communication-network reliability analysis: Probabilistic graph reduction,” in *Annual Reliability and Maintainability Symposium*, 1992.

[7] G. Lingner, T. Politof, and A. Satyanarayana, “A forbidden minor characterization and reliability of a class of partial 4-trees,” *Networks*, vol. 25, pp. 139–146, 1995.

[8] 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*, vol. 41, no. 4, pp. 512–517, Dec. 1992.

[9] M. Chari, T. Feo, and J. S. Provan, “The delta-wye approximation procedure for two-terminal reliability,” *Operations Research*, vol. 44, no. 5, pp. 745–757, 1996.

[10] E. Neufeld and C. Colbourn, “The most reliable series-parallel networks,” *Networks*, vol. 15, pp. 27–32, 1985.

[11] R. Karp and M. Luby, “Monte Carlo algorithms for planar multiterminal network reliability problems,” *Journal of Complexity*, vol. 1, no. 1, pp. 45–64, 1977.

[12] D. Karger, “A randomized fully polynomial time approximation scheme for the all terminal reliability problem,” *SIAM Review*, vol. 43, no. 3, pp. 499–522, 2001.

[13] G. Fishman, “A comparison of four Monte Carlo methods for estimating the probability of $s-t$ connectedness,” *IEEE Trans. Reliability*, vol. 35, no. 2, pp. 145–155, 1986.

[14] 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*, vol. 20, 1989.

[15] M. Ball, “Complexity of network reliability computations,” *Networks*, vol. 10, pp. 153–165, 1980.

[16] T. Brecht, “Lower bounds for two-terminal network reliability,” Master’s thesis, University of Waterloo, Waterloo, ON, 1985.

[17] R. Van Sylke and H. Frank, “Network reliability analysis: Part i,” *Networks*, vol. 1, pp. 279–290, 1972.

[18] H. AboElFotoh and C. Colbourn, “Computing 2-terminal reliability for radio-broadcast networks,” *IEEE Trans. Reliability*, vol. 38, no. 5, pp. 538–555, Dec. 1989.

[19] C. Colbourn and D. Harms, “Bounding all-terminal reliability in computer networks,” *Networks*, vol. 18, pp. 1–12, 1988.

[20] A. Ramanathan and C. Colbourn, “Counting almost minimum cutsets with reliability applications,” *Mathematical Programming*, vol. 39, pp. 253–261, 1987.

[21] L. Nel and C. Colbourn, “Combining Monte Carlo estimates and bounds for network reliability,” *Networks*, vol. 20, pp. 277–298, 1990.

[22] M. Ball and J. S. Provan, “Bounds on the reliability polynomial for shellable independence systems,” *SIAM J. Alg. Disc. Meth.*, vol. 3, pp. 166–181, 1982.

[23] K. Lee, H.-W. Lee, and E. Modiano, “Reliability in layered networks with random link failures,” MIT, Tech. Rep., 2009. [Online]. Available: http://web.mit.edu/kylee/www/reliability.pdf

[24] K. Lee and E. Modiano, “Cross-layer survivability in WDM-based networks,” in *IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009.

[25] F. Boesch, “On unreliability polynomials and graph connectivity in reliable network synthesis,” *Journal of Graph Theory*, vol. 10, no. 3, pp. 339–352, 1986.

[26] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge University Press, 1995.

[27] R. Karp and M. Luby, “Monte-carlo algorithms for enumeration and reliability problems,” in *Ann. Symp. on Found. of Comp. Sci.*, Nov. 1983.