# A Robust Optimization Approach to Network Design

by

Matthew R. Johnston

B.S. Electrical Engineering and Computer Science
University of California, Berkeley (2008)

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

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2010

Author ..........................................................................
Department of Electrical Engineering and Computer Science
July 28, 2010

Certified by ..........................................................................
Eytan Modiano
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by ..........................................................................
Professor Terry P. Orlando
Chairman, Committee on Graduate Students
Department of Electrical Engineering and Computer Science

# A Robust Optimization Approach to Network Design

by

Matthew R. Johnston

Submitted to the Department of Electrical Engineering and Computer Science
on July 28, 2010, in partial fulfillment of the
requirements for the degree of
Master of Science

## Abstract

This thesis addresses the problem of logical topology design for optical backbone networks subject to traffic following a Gaussian distribution. The network design problem is broken into three tasks: traffic routing, capacity allocation, and link placement. The routing and capacity allocation problems are formulated as a convex mathematical program. To extend this formulation to discrete optimization problems, such as the link placement sub-problem, it is reformulated as a mixed integer linear program (MILP) by extending tools from robust optimization to Gaussian variables. Bounds are presented to relate capacity allocation to the probability of traffic overflow on a link. Lastly, the link placement subproblem is formulated as an MILP and network topologies for deterministic traffic are compared with those for stochastic traffic.

Additionally, this thesis presents a scheme in which a dedicated backup network is designed to provide protection from random link failures. Upon a link failure in the primary network, traffic is rerouted through a preplanned path in the backup network. We introduce a novel approach for dealing with random link failures, in which probabilistic survivability guarantees are provided to limit capacity over-provisioning. We show that the optimal backup routing strategy in this respect depends on the reliability of the primary network. Specifically, as primary links become less likely to fail, the optimal backup networks employ more resource sharing amongst backup paths. We apply results from the field of robust optimization to formulate an ILP for the design and capacity provisioning of these backup networks. We then propose a simulated annealing heuristic to solve this problem for large-scale networks, and we present simulation results to verify our analysis on optimal backup networks.

Thesis Supervisor: Eytan Modiano
Title: Associate Professor of Aeronautics and Astronautics

# Acknowledgments

I am deeply grateful to Professor Eytan Modiano for providing a stimulating research environment, and offering endless guidance and support. Additionally, I would like to thank Dr. Hyang-Won Lee, whose insight was invaluable in this research process.

I would also like to thank my friends and family for their continued support throughout my academic career.

Lastly, this work was funded by NSF grant CNS-0626781 and by DTRA grant HDTRA1-07-1-0004.

Matt Johnston
Cambridge, MA

July 28, 2010

# Contents

# List of Figures

# Chapter 1

# Introduction

All-optical networks utilizing wavelength division multiplexing (WDM) are desirable for nationwide or global backbone networks. In such a network, nodes are connected by optical fibers with an enormous bandwidth. This bandwidth is broken down into multiple channels on different wavelengths, each operating at a bit rates of exceeding 10 Gbit/sec [11]. These high-speed networks are well-suited to handle the large information flow required in next-generation backbone networks, as internet traffic flow is doubling every year [24].

## 1.1 Optical Networks

The physical topology of the optical network is the collection of nodes and the fiber-optic links that connect those nodes. Nodes are equipped with wavelength routers to send packets from one wavelength on an incoming fiber to some wavelength on an outgoing fiber, without converting the packets from the optical domain to the electrical domain. By configuring these routers, a path is created traversing a wavelength over one or more fiber links, from one end-node to another. These transparent virtual links are referred to as lightpaths. To connect lightpaths to one another, nodes also contain an electronic switch to convert the information flow to the electronic domain, process it, and return it to the optical domain. Multiple lightpaths can traverse the same physical link, but these lightpaths must operate on separate wavelengths to keep the traffic flows independent.

In addition to the physical topology, a logical topology is made up of the interconnection of nodes with opto-electronic switches and lightpaths. In the logical topology, each link corresponds to a lightpath possibly spanning multiple physical fibers. Traffic through the WDM network is routed over the logical topology, due to the ease of transmitting traffic over lightpaths. The size of the logical topology is restricted by the limited processing capability of electronic switches, as only a few lightpaths can be set up to arrive and/or terminate at a node [18]. Aside from this logical degree constraint, there is a limit on the number of wavelengths on each optical fiber, dictated by the bandwidth of the cable and the desired bit rate of the channel. Consequently, a fully-connected logical mesh is generally infeasible, and a partially connected topology must be designed such that the traffic can still be routed effectively through the network.

## 1.2 Logical Topology Design for Stochastic Demands

The problem of designing a logical topology over an existing physical topology has been studied extensively [17, 18, 29]. The objective is to place logical links on the physical topology, assign capacity to these links, and route the traffic over the logical topology. The physical topology is known a priori, as is a matrix describing the long-term average traffic flow between any node-pair in the network. Capacity is assigned to the logical links such that the traffic routed on each link will not exceed its capacity. Networks are commonly designed to either minimize the maximum link capacity in the network or the total capacity used over all links in the network. Minimizing the maximum link capacity is equivalent to minimizing the network congestion, or balancing the traffic load throughout the network. This approach mitigates the effect of bottlenecks on network performance. Additionally, minimizing total capacity may result in solutions with one or more heavily capacitated links relative to others, and the network may not have the resources to support such configurations. For these reasons, the primary focus is minimizing the maximum link capacity; however, most of the formulations throughout this thesis can be readily extended to the case of minimizing total capacity.

Some authors use alternate objective functions for the design formulation [18]. For example, one can consider a solution achieving a minimal number of hops for the paths used to route demands, since at each hop, an opto-electronic conversion of the packet flow is required, which can lead to delays. However, the minimization of the average weighted number of hops leads to solutions with minimal average traffic flowing on each link, and is therefore strongly related to minimizing the maximum congestion level. Alternatively, formulations can minimize average packet delay in the network. Since wide area networks are considered, propagation delay dominates queuing delay unless the link load is very close to the limit enforced by the channel capacity. Minimizing propagation delay is a shortest-path routing solution, where minimizing queuing delay is achieved indirectly by minimizing congestion, as both solutions will result in low average link traffic. Consequently, we focus on minimizing network congestion.

### 1.2.1 Logical Topology Design Formulation

The logical topology design problem is formulated as a mixed-integer linear program (MILP). Let $\Lambda$ be the traffic matrix, so that $(\Lambda)_{sd} = \lambda^{sd}$ is the traffic demand from node $s$ to node $d$. The demand $\lambda^{sd}$ is a deterministic quantity representing the average (mean) long term traffic flow between two nodes. Let $C_{ij}$ represent the capacity allocated to link $(i, j)$, and $\lambda_{ij}^{sd}$ be the traffic from $s$ to $d$ that traverses link $(s, d)$. Additionally, let $b_{ij}$ be binary design variables which will be equal to 1 if and only if a directed link is to be placed from node $i$ to node $j$. The MILP for network design can be formulated as follows:

$$\text{Minimize:} \quad C_{\max} \tag{1.1}$$

$$\text{Subject to:} \quad C_{\mathrm{ij}} \geq \sum_{sd} \lambda_{ij}^{sd} \qquad \forall (i,j) \tag{1.2}$$

$$\sum_{j} \lambda_{ij}^{sd} - \sum_{j} \lambda_{ji}^{sd} = \begin{cases} \lambda^{sd}, & \text{if } s = i \\ -\lambda^{sd}, & \text{if } d = i \\ 0, & \text{otherwise} \end{cases} \qquad \forall s, d, i \tag{1.3}$$

$$C_{ij} \leq C_{\max} \qquad \forall i, j \tag{1.4}$$

$$\lambda_{ij}^{sd} \leq b_{ij} \lambda^{sd} \qquad \forall s, d, i, j \tag{1.5}$$

$$\sum_{i} b_{ij} = \Delta_i \qquad \forall j \tag{1.6}$$

$$\sum_{j} b_{ij} = \Delta_o \qquad \forall i \tag{1.7}$$

$$\lambda_{ij}^{sd} \geq 0 \qquad \forall s, d, i, j \tag{1.8}$$

$$b_{ij} \in \{0, 1\} \qquad \forall i, j \tag{1.9}$$

The MILP presented above minimizes the maximum link capacity, but (1.1) could be replaced with $\sum_{ij} C_{ij}$ to minimize total capacity instead. Constraint (1.2) allocates capacity to support the total traffic on each link. Equation (1.3) is a general flow conservation constraint, enforcing the flow into each node be equal to the flow out of that node, with exceptions for the source and destinations of a demand. Constraint (1.5) restricts traffic to only flow between nodes where links have been placed. The link placements themselves are restricted by (1.6) and (1.7), where $\Delta_i$ and $\Delta_o$ are bounds on the logical node degree.

For simplicity, the wavelength restrictions on each physical link are ignored in this formulation, as we assume that there are sufficiently many wavelengths available on each physical fiber [29]. Additionally, wavelength converters allow lightpaths to be constructed over different wavelengths on different fibers. Thus, the physical topology does not appear in this formulation, since there are no delay constraints or wavelength constraints.

This formulation allows traffic between any source-destination pair to be bifurcated among different paths in the network. Consequently, there is no limitation on the number of paths a demand can take. Additionally, at most one lightpath can be placed between any two nodes in the network.

The MILP formulated can be solved using commercially available solvers such as CPLEX; however, this problem is known to be difficult to solve for large networks. Computation time depends on the design constraints used and the number of these constraints. Heuristics are commonly used in order to solve this MILP for larger networks [18, 29].

### 1.2.2 Motivation for Stochastic Demands

The design problem above relies on the availability of the traffic matrix $\Lambda$, containing the average traffic flow between each node pair. However, these traffic matrices are commonly unavailable since direct measurements are impractical. Estimation techniques are used to obtain this data from link load measurements, and the traffic intensities coming from these estimations are prone to errors. Additionally, the design problem is typically solved during the network configuration stage, where the exact traffic information is unavailable.

Traffic demands are also becoming harder to predict, and tend to change with time. As new services utilizing optical backbone networks are created and removed, demands between nodes fluctuate. Consequently, a network design based off of a traffic matrix estimated from today's link measurements may not be suitable to carry demand in the future. Furthermore, network failures and service disruptions also cause variations in traffic demands.

Due to the uncertain nature of the network traffic, network links must be provisioned with sufficient capacity to support the possible traffic fluctuations. Demands are represented as random variables rather than deterministic quantities to capture the traffic variability. When treating node demands as random variables, there is a significant probability that the capacity allocated will be underutilized. Furthermore, traffic loads may have a very large or no upper bound, and it becomes impractical to allocate capacity to support all the possible traffic realizations. Therefore, we provide a framework in which link capacity is allocated such that traffic is supported with some high probability, rather than complete certainty.

The volatility of random variables is described by their standard deviations. A demand with a high standard deviation requires a large capacity to support, but it is likely that the capacity will be underutilized. Consequently, the demands should be routed to minimize the traffic variability. As an example, consider two independent and identically distributed (i.i.d) random variables $X$ and $Y$ with $E[X] = E[Y] = \mu$ and $\text{var}(X) = \text{var}(Y) = \sigma^2$. Now, consider the sum, $Z = X + Y$. The new random variable $Z$ has expectation $2\mu$ and standard deviation $\sigma\sqrt{2} < 2\sigma$. As a result, assigning capacity to support $X$ and $Y$ separately requires more capacity than required to support the sum $Z$.

This suggests a fundamental difference between routing deterministic traffic and stochastic traffic. Furthermore, this difference in the routing sub-problem suggests that the optimal link placements for stochastic traffic will differ from those for deterministic demands. In this work, we reformulate the entire logical topology design problem to place links, allocate capacity, and route traffic to support stochastic traffic with high probability and the lowest possible cost.

### 1.2.3 Robust Optimization

A natural technique to apply to optimization problems with data variability is robust optimization. Robust optimization is a method of finding a solution to a problem that best fits all possible realizations of data subject to uncertainty [9]. The first approach to robust optimization was taken by Soyster [32]. This work proposes a linear program to construct a solution that is feasible for all possible realizations of a bounded demand. These models produce solutions that are very conservative. In the robust optimization literature, conservatism refers to a solution that is unnecessarily robust. Ben-Tal and Nemirovski [3] address the issue of over-conservatism in robust optimization formulations by proposing less-conservative solutions to linear problems with ellipsoidal uncertainties. This approach relies on nonlinear, but convex, formulations which do not extend easily to discrete optimization problems.

Lastly, Bertsimas and Sim [4] propose a linear formulation with an adjustable level of robustness. The optimization parameters are uncertain, but have some unknown symmetric distribution taking values entirely in a finite interval. For each constraint $i$, a parameter $\Gamma_i$ is introduced. If fewer than a predefined number of coefficients $\Gamma_i$ change, the solution is guaranteed to be feasible. On the other hand, if more coefficients change, the solution is still feasible with high probability. Bertsimas and Sim provide tight probability bounds on symmetric, bounded random variables in order to quantize this probability. By changing the value of $\Gamma_i$, the authors can trade off between the level of robustness and conservatism.

### 1.2.4 Our Contributions

In this work, we formulate and analyze the problem of logical topology design under Gaussian-distributed traffic. First, we use tools from stochastic programming to formulate a second order cone programming problem to solve the routing and capacity allocation sub-problems over a given network. Then an approach using dedicated support for each demand is proposed, which is shown to be linear, but much more conservative than the stochastic programming approach. We then extend the work of Bertsimas and Sim in [4] to Gaussian random variables, by formulating a robust optimization problem and developing new bounds on the probability of capacity constraint violation. This leads to MILP formulation to route traffic and allocate capacity, for which we propose heuristics to solve.

To complete the topology design problem, we then consider the link-placement sub-problem in addition to the routing and capacity allocation sub-problems. Using the robust optimization techniques described above, we formulate the design problem as a MILP and propose heuristic algorithms to solve it. Lastly, we explore some properties of network design under stochastic traffic demands, and how the problem differs from its deterministic counterpart.

### 1.2.5 Previous Work

Several other authors have examined the problem of routing and capacity allocation with traffic uncertainty. Most works in this area use a bounded model for the traffic distribution [2, 15]. These models are referred to as polyhedral models, in which the traffic from different sources is bounded and satisfies various linear constraints. A specific example is the "hose" model, which bounds the sum of demands originating and ending at a given node.

Additionally, [15] presents a traffic model in which the demand between each node pair is restricted to an interval. In this case, the results of [4] are used to create a robust optimization formulation for routing and capacity allocation. We extend this work by considering Gaussian traffic, which is unbounded in nature. Additionally, we explore the problem of link placement in addition to routing and capacity allocation for stochastic traffic which was neglected in these works.

Some authors have formulated related network design problems for Gaussian traffic. The authors in [10] implement a tabu-search heuristic to design networks based on normally distributed demands, but do not attempt to formulate the problem mathematically. In contrast, [22] formulate an offline traffic engineering problem based off of Gaussian traffic. They argue that the objective should be a weighted combination of maximizing revenue while minimizing risk. Here, risk is the potentially underutilized capacity that is provisioned to the network. By including this in the objective, they prevent over-provisioning of capacity. In our work, we instead consider a fixed probability of traffic exceeding the provisioned capacity on each link. This is an alternate of reducing over-provisioning in the presence of stochastic traffic.

The works in [21, 35] are most relevant to our work. In [21], the network routing and allocation problems are formulated assuming stochastic demands by changing probabilistic constraints to linear, deterministic constraints. While this work also considers a probability of constraint violation, the method proposed results in over-provisioning capacity. On the other hand, [35] formulates the optimal stochastic formulation for gaussian traffic routing and allocation; however, their formulation is non-linear which prevents this approach from being applied to discrete optimization problems, such as link placement. Our work compares both these formulations and uses tools from robust optimization to formulate a linear problem that is less-conservative as the formulation in [21].

The link placement sub problem with stochastic traffic has only been explored in a few other works. In [16, 39], a design and routing scheme is proposed that is insensitive to the traffic matrix. The backbone network is designed to perform equally well for all valid traffic matrices, as it is too hard to predict the exact value of the traffic demands. In [39], the approach is referred to as Valiant Load-balancing (VLB). The VLB approach establishes a fully-connected logical network, then routes traffic evenly across all two-hop paths. Based on some upper bounds on the traffic matrix, capacity is then allocated to each logical link such that any realization of the traffic matrix can be supported.

These approaches completely ignore the distribution of the random demands, resulting in over-provisioning capacity. In our work, by allowing a small probability that the traffic cannot be supported, we can support unbounded traffic without significant over-provisioning. We argue that characteristics of the distribution of traffic should be taken into account to optimally determine the correct routing strategy.

## 1.3 Backup Network Design for Survivability Against Random Network Failures

Today's backbone networks are designed to operate at very high data rates, now exceeding 10 Gbit/s [11]. Consequently, any link failure can lead to catastrophic data loss. In order to ensure fast recovery from failures, protection resources must be allocated prior to any network failures.

### 1.3.1 Spare Capacity Allocation

A widely used approach for recovery from a link failure is *preplanned link restoration* [20], where a backup path between the end nodes of a link is chosen for every link during the network configuration stage. In the event of a link failure, the disrupted traffic can be rerouted onto its backup path. Preplanned methods of link restoration offer benefits over other methods in terms of speed and simplicity of failure recovery, as no additional dynamic routing is necessary at the time of a failure [40]. In addition to designing a backup path for each link, preplanned link restoration requires provisioning of sufficient spare capacity along each backup path to carry the load of failed links. Backup paths can share spare capacity and network resources to reduce the total cost of protection.

Spare capacity allocation for link-based protection has been studied extensively in the context of single-link failures [11, 19, 28, 37]. The objective of these works is to allocate sufficient protection resources to recover from any single link failure. Recently, the authors in [1] proposed the use of a dedicated backup network to protect against a single failure on the primary network. Upon such a failure, the load on the failed link is routed on a predetermined path on the backup network. The authors provide an Integer Linear Program (ILP) to design an optimal backup network with minimal cost. They show that the cost of the optimal backup network is small relative to that of a large primary network. Specifically, they show that the ratio between the total backup capacity and the total primary capacity tends to zero as the network size grows large for certain classes of networks.

### 1.3.2 Robustness to Multiple Failures

Communication networks can suffer from multiple simultaneous failures, for example, if a second link fails before a first failed link is repaired. Furthermore, natural disasters or large scale attacks can destroy several links in the vicinity of such events. Preplanning backup paths for combinations of multiple failures can be complex and impractical, and can lead to significant capacity over-provisioning. Consequently, new approaches must be considered to offer protection against multiple failures.

Several authors have extended the results of survivability for single link failures to dual-link failures [20, 8, 12]. The work in [14] considers protecting against up to three link failures. Most of these works require the primary network to have multiple disjoint paths between node pairs to survive multiple failures. However, this assumption is too restrictive when considering a large number of failures. Additionally, [13] provides a spare capacity allocation approach based on a specific set of failure events, and restricted backup path lengths. In all of these works, large amounts of spare capacity are required if many links can fail simultaneously.

Survivability amidst multiple failures has also been addressed in the form of a Shared Risk Link Group (SRLG) [26]. An SRLG is a set of links sharing a common network resource, such that a failure of that resource could lead to a failure of all links in the SRLG. Many authors have proposed routing strategies for path-based protection against SRLG failures [5, 23, 25, 36]. These works assume that links in a SRLG all fail simultaneously and deterministically. However, this line of work does not extend to independent, random failures.

### 1.3.3 Our Contributions

We introduce a new framework for providing protection from multiple random link failures involving probabilistic survivability guarantees. Since large-scale attacks and natural disasters can result in multiple links failing randomly, providing protection from any single failure is insufficient, and networks designed for protection against single-link failures often cannot protect against multiple failures. The straight-forward approach of offering guaranteed protection against any random failure scenario is to allocate capacity such that every failure event is protected. However, this approach is impractical as it requires enormous amounts of capacity to protect against potentially unlikely events. By allocating capacity to offer protection with high probability, the total cost of protection is greatly reduced.

Motivated by [1] and the simplicity of their approach, we extend the use of a dedicated backup network to deal with multiple random link failures. We show that a dedicated backup network is a low-cost method of providing protection against random failures, relative to large primary networks. Additionally, we show that the structure of the minimum-cost backup network changes with the reliability of the primary network. Specifically, optimal backup networks for primary networks with a low link-failure probability employ a high level of link sharing amongst backup paths. On the other hand, optimal backup networks for primary networks with a high link-failure probability emphasize shorter backup paths, and less capacity sharing.

To design a backup network under random link failures, we develop a robust optimization approach to spare capacity provisioning. Robust optimization finds a solution that is robust to uncertainty in the optimization parameters [3, 4, 32]. In [4], Bertsimas and Sim propose a novel linear formulation with an adjustable level of robustness. These techniques have been successfully applied to network flow problems [9]. We apply these results to design backup networks robust to the uncertainty in link failures, leading to an ILP formulation for backup capacity provisioning. We also present a simulated annealing approach to solve the ILP for large-scale networks.

Chapter 2

# Network Design for Stochastic Traffic

As mentioned in Chapter 1, the network topology must be robust to traffic variations. Our approach is to model the traffic demands as random variables with known distributions. We reformulate the mixed-integer linear program for logical topology design presented in 1.2.1 for random demands as a stochastic optimization problem.

## 2.1 Stochastic Traffic Matrix

The problem of inferring source-destination traffic intensities from aggregated link traffic measurements is referred to as network tomography. In the network tomography literature, demands between node-pairs have been described using a Gaussian distribution [7]. Intuitively, many independent sources contribute to each demand, and by the Central Limit Theorem, the resulting distribution of the traffic can be approximated by a Gaussian distribution. Therefore, each entry of the traffic matrix is modeled as an independent Gaussian random variable with mean $\mu_{sd}$ and variance $\sigma^2_{sd}$.

$$\lambda^{sd} \sim \mathcal{N}(\mu_{sd}, \sigma^2_{sd}) \tag{2.1}$$

The traffic from a source to a destination should be nonnegative in order to accurately model traffic flow intensity; however, the probability density function (PDF) of a Gaussian random variable is not restricted to positive values. Demands can be modeled as truncated normal random variables instead, truncated at zero . This would be necessary for demands with a small mean and high variance. In this work, it is assumed that $\sigma_{sd}$ is small relative to the mean such that the event that $\lambda^{sd}$ is negative is negligible.

## 2.2 Stochastic Programming Formulation

Since the demands between each node pair are represented as random variables and the exact traffic cannot be specified, new routing variables are introduced to replaced the $\lambda_{ij}^{sd}$'s in the formulation of Section 1.2.1. Let $a_{ij}^{sd}$ represent the fraction of the traffic flow from $s$ to $d$ traversing link $(i,j)$. In other words,

$$\lambda_{ij}^{sd} = a_{ij}^{sd}\lambda^{sd} \tag{2.2}$$

where $0 \leq a_{ij}^{sd} \leq 1 \quad \forall(s,d),(i,j)$. The MILP in section 1.2.1 is rewritten using equation 2.2 as shown below.

$$\begin{aligned}
\text{Minimize:} \quad & C_{\max} \\
\text{Subject to:} \quad & C_{\mathrm{ij}} \geq \sum_{sd} a_{ij}^{sd}\lambda^{sd} \qquad \forall(i,j) && (2.3) \\
& \sum_j a_{ij}^{sd} - \sum_j a_{ji}^{sd} = \begin{cases} 1, & \text{if } s = i \\ -1, & \text{if } d = i \\ 0, & \text{otherwise} \end{cases} \qquad \forall s,d,i && (2.4) \\
& C_{ij} \leq C_{\max} \qquad \forall i,j \\
& a_{ij}^{sd} \leq b_{ij} \qquad \forall s,d,i,j && (2.5) \\
& \sum_i b_{ij} = \Delta_i \qquad \forall j \\
& \sum_j b_{ij} = \Delta_o \qquad \forall i \\
& a_{ij}^{sd} \geq 0 \qquad \forall s,d,i,j \\
& b_{ij} \in \{0,1\} \qquad \forall i,j
\end{aligned}$$

Notice the random variables $\lambda^{sd}$ appear only in the capacity constraint in (2.3). Since the $\lambda^{sd}$ are unbounded, a finite capacity $C_{ij}$ cannot be specified such that (2.3) is satisfied with probability 1. Therefore, we instead consider a probabilistic alternative:

$$\mathbf{P}\left(\sum_{sd} a_{ij}^{sd}\lambda^{sd} > C_{ij}\right) \leq \epsilon \qquad \forall(i,j) \tag{2.6}$$

In the above equation, $\epsilon$ represents the probability of capacity constraint violation. This occurs when the link traffic exceeds the capacity allocated to that link. To incorporate this constraint into a mathematical formulation, it needs to be converted from a probabilistic constraint to a deterministic one. Since for each $(s,d)$, $\lambda^{sd}$ is an independent Gaussian random variable, the link traffic, a weighted sum of independent Gaussians, is also Gaussian with mean $\sum_{sd} a_{ij}^{sd}\mu_{sd}$ and variance $\sum_{sd}(a_{ij}^{sd}\sigma_{sd})^2$. It is then straightforward to convert (2.6) to a deterministic constraint in terms of the CDF of a normal random variable, $\Phi()$.

$$\mathbf{P}\left(\sum_{sd} a_{ij}^{sd}\lambda^{sd} > C_{ij}\right) \leq \epsilon \qquad \forall(i,j)$$

$$1 - \Phi\left(\frac{C_{ij} - \sum_{sd} a_{ij}^{sd}\mu_{sd}}{\sqrt{\sum_{sd}(a_{ij}^{sd}\sigma_{sd})^2}}\right) \leq \epsilon \qquad \forall(i,j) \tag{2.7}$$

$$C_{ij} \geq \sum_{sd} a_{ij}^{sd}\mu_{sd} + \Phi^{-1}(1-\epsilon)\sqrt{\sum_{sd}(a_{ij}^{sd}\sigma_{sd})^2} \qquad \forall(i,j) \tag{2.8}$$

Equation (2.8) is a non-linear, convex constraint for the capacity needed to support some routing of the traffic with a given probability $\epsilon$.

The deterministic formulation for topology design in Section 1.2.1 is an MILP, solvable by many commercially available optimization tools. However, replacing the constraint in (2.3) with inequality (2.8) introduces a non-linearity to the optimization problem. Mixed integer convex programming problems are not necessarily solvable, particularly by today's commercially available solvers. Therefore, it is preferable to have a linear approximation to (2.8) such that integer constraints can be added.

Therefore, we initially assume that the link placement problem has already been solved. Denote the set of links with $b_{ij} = 1$ as $\mathcal{L}$. The convex optimization formulation for routing and capacity allocation is as follows.

$$\begin{aligned}
\text{Minimize:} \quad & C_{\max} \\
\text{Subject to:} \quad & C_{ij} \geq \sum_{sd} a_{ij}^{sd}\mu_{sd} + \Phi^{-1}(1-\epsilon)\sqrt{\sum_{sd}(a_{ij}^{sd}\sigma_{sd})^2} \quad \forall (i,j) \in \mathcal{L} \\
& \sum_j a_{ij}^{sd} - \sum_j a_{ji}^{sd} = \begin{cases} 1, & \text{if } s = i \\ -1, & \text{if } d = i \qquad \forall s, d, i \\ 0, & \text{otherwise} \end{cases} \\
& C_{ij} \leq C_{\max} \qquad \forall (i,j) \in \mathcal{L} \\
& a_{ij}^{sd} \geq 0 \qquad \forall s, d, (i,j) \in \mathcal{L}
\end{aligned} \tag{2.9}$$

The above formulation is a second order cone programming problem (SOCP) that can be solved by convex and quadratic optimizers (e.g. LOQO). However, as the network size grows large, this optimization becomes difficult. For example, running formulation (2.9) on the network in Figure 2-4 requires a computation time that increases exponentially with the number of demands, as shown in Figure 2-1. Moreover, as stated previously, this convex formulation is not useful for discrete optimization problems, such as link placement.

## 2.3 Conservative Linear Formulation

In order to obtain a more tractable formulation, as well as one that can be extended to integer optimization problems, the mathematical program in (2.9) should be formulated as a linear program (LP). Suppose a tighter capacity constraint is used

$$\mathbf{P}(\lambda^{sd} > C^{sd}) \leq \epsilon \qquad \forall (s,d). \tag{2.10}$$

Constraint (2.10) restricts the probability that each demand exceeds its assigned

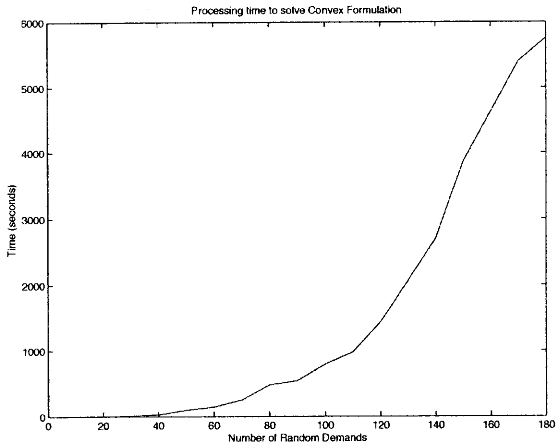


Figure 2-1: Processing Time for formulation (2.9) using the LOQO nonlinear optimization solver

capacity. By assigning dedicated capacity to each demand, statistical multiplexing gains are unachievable and capacity is over-provisioned. The probabilistic constraint in (2.10) can be converted to a deterministic constraint $C^{sd} \geq \mu_{sd} + \Phi^{-1}(1-\epsilon)\sigma_{sd}$, and link capacity constraint follows as

$$C_{ij} \geq \sum_{sd} a_{ij}^{sd}(\mu_{sd} + \Phi^{-1}(1-\epsilon)\sigma_{sd}) \qquad \forall (i,j) \in \mathcal{L}. \tag{2.11}$$

Constraint (2.11) leads to an LP formulation for routing and capacity allocation.

$$\begin{aligned}
\text{Minimize:} \quad & C_{\max} \\
\text{Subject To:} \quad & C_{ij} \geq \sum_{sd} a_{ij}^{sd}\mu_{sd} + \Phi^{-1}(1-\epsilon)\sum_{sd} a_{ij}^{sd}\sigma_{sd} \quad \forall (i,j) \in \mathcal{L} \\
& \sum_j a_{ij}^{sd} - \sum_j a_{ji}^{sd} = \begin{cases} 1, & \text{if } s = i \\ -1, & \text{if } d = i \qquad \forall s,d,i \\ 0, & \text{otherwise} \end{cases} \\
& C_{ij} \leq C_{\max} \qquad \forall (i,j) \in \mathcal{L} \\
& a_{ij}^{sd} \geq 0 \qquad \forall s,d,(i,j) \in \mathcal{L}
\end{aligned} \tag{2.12}$$

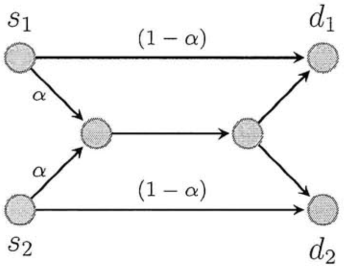


Figure 2-2: Example network, with i.i.d. demands $\lambda^{s_1 d_1}, \lambda^{s_2 d_2} \sim \mathcal{N}(\mu, \sigma)$.

The capacity allocated in (2.12) is always greater than or equal to the capacity provisioned in the solution to (2.9). Borrowing terminology from optimization literature, formulation (2.12) is referred to as *conservative*, because it allocates more than the minimum required capacity to maintain the probability of constraint violation. However, the linearity of (2.12) enables extension to discrete optimization problems.

### 2.3.1 Illustrative Examples

The dedicated capacity provisioning scheme is wasteful in terms of the maximum capacity on any link in the network. This over-provisioning can be reduced by adding post processing to recalculate the link capacities using the convex constraint (2.8); however, routing decisions are made using the sub-optimal linear capacity constraint. We study an illustrative example to compare the two routing strategies.

Consider the network shown in Figure 2-2. Traffic demands exist from source $s_1$ to destination $d_1$ and from $s_2$ to $d_2$, and are independently and identically distributed according to $\mathcal{N}(\mu, \sigma^2)$. Due to symmetry, both demands have the same fraction of traffic ($\alpha$) routed on the shared link. Clearly, the optimal solution allocates equal capacity on all links. Using the linear capacity constraint in (2.11), we can analytically solve for the optimal routing fraction $\alpha$. Let $k = \Phi^{-1}(1-\epsilon)$.

$$C_L = (1-\alpha)\mu + (1-\alpha)k\sigma = 2\alpha\mu + 2\alpha k\sigma \tag{2.13}$$

$$\alpha = \frac{1}{3} \tag{2.14}$$

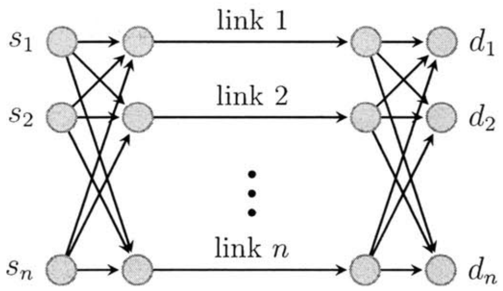


Figure 2-3: Example network, with i.i.d. demands $\lambda^{s_i d_i} \sim \mathcal{N}(\mu, \sigma)$. Each $(s, d)$ node pair has the option to send traffic over any of the links.

Additionally, the same can be done for the non-linear capacity constraint in (2.8).

$$C_{NL} = (1-\alpha)\mu + (1-\alpha)k\sigma = 2\alpha\mu + \sqrt{2}\alpha k\sigma \tag{2.15}$$

$$\alpha = \frac{\mu + k\sigma}{3\mu + (1+\sqrt{2})k\sigma} \tag{2.16}$$

In contrast to the optimal routing for stochastic traffic, the linear formulation routes traffic as if it were deterministic. Conversely, for traffic with positive variance, more than 1/3 of the traffic is routed over the shared link, since the traffic on the shared link has a smaller standard deviation than the dedicated link traffic. As the ratio $\frac{\sigma}{\mu}$ increases, a larger fraction of traffic is routed on the shared link.

As a second example, consider the network in Figure 2-3 with $n$ demands, each going from $s_i$ to $d_i$. An optimal routing for the dedicated capacity provisioning formulation in (2.11) is to send the traffic from $s_i$ to $d_i$ entirely over link $i$. Other solutions exist for this formulation, but we will focus on this routing for comparison. Using this strategy, the maximum link capacity on any of the links is $\mu + k\sigma$.

The solution to the convex formulation routes demand equally over each link. In this case, the capacity on each link is

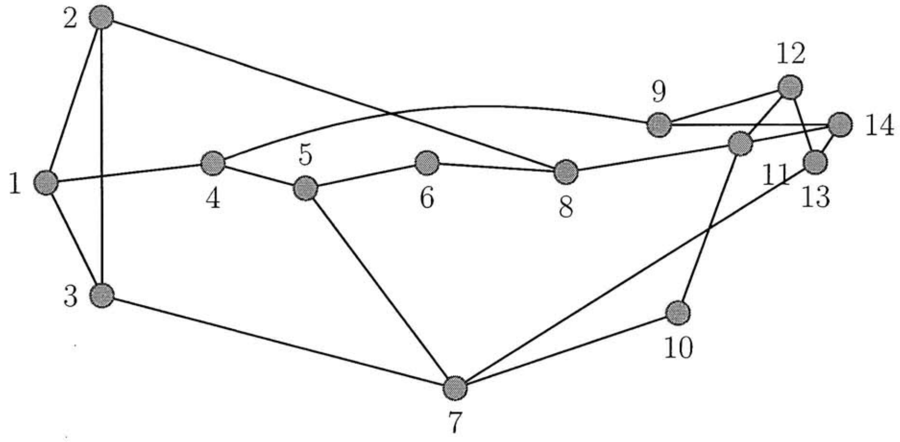


Figure 2-4: 14 Node NSFNET backbone network (1991)

$$C_i = \sum_{i=1}^{n} \frac{\mu}{n} + k\sqrt{\sum_{i=0}^{n} \frac{\sigma^2}{n^2}} \tag{2.17}$$

$$= \mu + \frac{k\sigma}{\sqrt{n}} \tag{2.18}$$

In the large $k\sigma$ regime, the stochastic formulation results in a max capacity that is a fraction $1/\sqrt{n}$ of that required by the linear formulation. In the limiting case as $n$ grows large, the required link capacity is the mean $\mu$ of a traffic demand, whereas a capacity of $\mu + k\sigma$ is required on each link for the linear approach.

As a last example, we consider the real-world NSFNET shown in Figure 2-4. Demands, which satisfy $\lambda^{s_i d_i} \sim \mathcal{N}(\mu, \sigma^2)$, exist between random source and destination pairs. All links on the NSFNET graph are assumed to be bidirectional. For each random set of N demands, both the LP of (2.12) and the optimal convex optimization of (2.9) are solved, and the results are plotted in figure 2-5. As can be seen, the optimal routing and capacity allocation offers a savings of 30% in maximum link capacity over the linear approach.

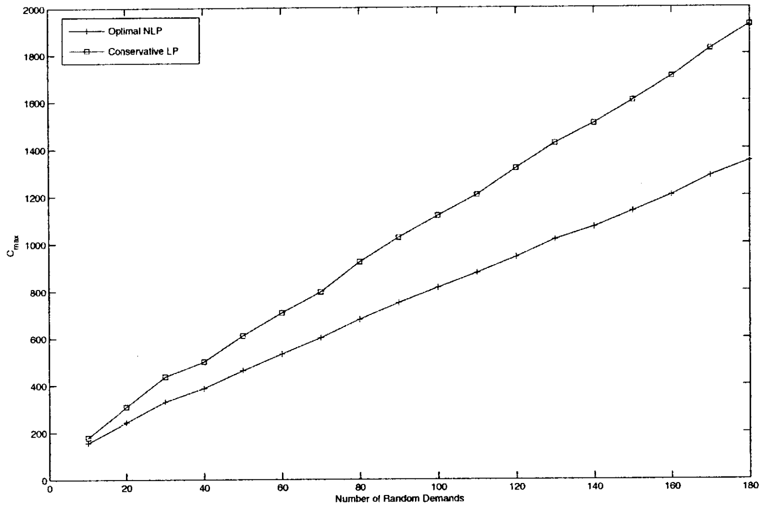


Figure 2-5: Optimal values of the NLP in (2.9) and the LP in (2.12) for the NSFNET in Figure 2-4. The x-axis shows the number of random demands generated for the network. The LP was solved using CPLEX, and the non-linear program (NLP) was solved using LOQO

## 2.4 Robust Optimization

In Section 2.2, a convex optimization problem was formulated to compute the optimal routing and capacity allocation over a network to support Gaussian-distributed demands with a given probability of error. To extend the formulation to discrete optimization problems, a conservative LP formulation was proposed. While this formulation is linear, it is wasteful in terms of allocated capacity. Results from robust optimization can be used to formulate an LP for routing and capacity allocation with less capacity over-provisioning.

### 2.4.1 The Bertsimas and Sim Formulation

As explained in Section 1.2.3, Bertsimas and Sim propose a linear programming approach to robust optimization with an adjustable level of robustness [4]. They assume the demands $\lambda^{sd}$ are symmetric random variables bounded within $[\bar{\lambda}^{sd} - \hat{\lambda}^{sd}, \bar{\lambda}^{sd} + \hat{\lambda}^{sd}]$, where $\bar{\lambda}^{sd} = \mathbf{E}[\lambda^{sd}]$. For each $(i, j)$, a parameter $\Gamma_{ij}$ is introduced taking a value between 0 and $n_{ij}$, the number of demands traversing link $(i, j)$. Their approach is to assign capacity to protect against any scenario where $\Gamma_{ij}$ of the demands exceed their mean, by adding capacity to support the worst-case realization of those random variables. The other demands are assumed not to exceed their mean, and are therefore allocated a capacity of $\bar{\lambda}^{sd}$. This approach has the property that if more than $\Gamma_{ij}$ demands exceed their mean, the robust solution is still feasible with high probability. Consequently, varying the parameter $\Gamma_{ij}$ adjusts the robustness of the formulation. For $\Gamma_{ij}$ close to 0, little capacity is allocated and the probability this capacity is insufficient to support the traffic is high. Conversely, setting $\Gamma_{ij}$ large results in over-provisioned link capacity.

We apply the Bertsimas and Sim approach to our problem. Initially, assume demands are drawn from a truncated normal distribution, i.e. the traffic $\lambda_B^{sd}$ between demand pair $(s, d)$ satisfies $\lambda_B^{sd} \sim \mathcal{N}(z | z \leq \mu_{sd} + k\sigma_{sd}; \mu_{sd}, \sigma_{sd}^2)$ for some constant $k$, and has a PDF

$$f_{\lambda_B^{sd}}(z; \mu_{sd}, \sigma_{sd}^2) = \frac{\frac{1}{\sigma_{sd}}\phi(\frac{z-\mu_{sd}}{\sigma_{sd}})}{\Phi(k)} \qquad z < \mu_{sd} + k\sigma_{sd} \tag{2.19}$$

where $\phi()$ and $\Phi()$ are the PDF and CDF of a standard normal random variable respectively.

We formulate a non-linear program according to [4]. Let $\mathcal{D}$ be the set of all demands. Let $S_{ij} \subseteq \mathcal{D}$ be a subset of demands restricted to be of size $\lfloor \Gamma_{ij} \rfloor$, and let $t_{ij} \in \mathcal{D} \setminus S_{ij}$ be another demand that is not in $S_{ij}$. On each link $(i, j)$, sufficient capacity is allocated to support $a_{ij}^{sd}\mu_{sd}$ for each demand, as well as the $\Gamma_{ij}$ largest values of $a_{ij}^{sd}k\sigma_{sd}$. In the case where $\Gamma_{ij}$ is not an integer, the fraction $(\Gamma_{ij} - \lfloor \Gamma_{ij} \rfloor)$ of $a_{ij}^{t_{ij}}k\sigma_{t_{ij}}$ is also supported. The formulation is presented below.

$$
\begin{aligned}
\text{min.} \quad & C_{\max} \\
\text{s.t.} \quad & C_{ij} \geq \sum_{sd \in \mathcal{D}} \mu_{sd} a_{ij}^{sd} + \\
& \max_{S_{ij} \cup \{t_{ij}\} | S_{ij} \in \mathcal{D}, |S_{ij}| = \lfloor \Gamma_{ij} \rfloor, t_{ij} \in \mathcal{D} \setminus S_{ij}} \left\{ \sum_{(s,d) \in S_{ij}} k\sigma_{sd} a_{ij}^{sd} + (\Gamma_{ij} - \lfloor \Gamma_{ij} \rfloor) k\sigma_{t_{ij}} a_{ij}^{t_{ij}} \right\} \quad \forall (i,j) \\
& \sum_j a_{ij}^{sd} - \sum_j a_{ji}^{sd} = \begin{cases} 1, & \text{if } s = i \\ -1, & \text{if } d = i \\ 0, & \text{otherwise} \end{cases} \quad \forall (s,d), i \\
& C_{ij} \leq C_{max} \qquad \forall (i,j) \\
& a_{ij}^{sd} \geq 0 \quad \forall i, j, s, d
\end{aligned}
\tag{2.20}
$$

Formulation (2.20) is not linear, but can be reformulated as an LP by following the procedure used in [4]. For a vector $\mathbf{a_{ij}}$, consider the function

$$
\beta_{ij}(\mathbf{a_{ij}}, \Gamma_{ij}) = \max_{S_{ij} \cup \{t_{ij}\} | S_{ij} \in \mathcal{D}, |S_{ij}| = \lfloor \Gamma_{ij} \rfloor, t_{ij} \in \mathcal{D} \setminus S_{ij}} \left\{ \sum_{(s,d) \in S_{ij}} k\sigma_{sd} a_{ij}^{sd} + (\Gamma_{ij} - \lfloor \Gamma_{ij} \rfloor) k\sigma_{t_{ij}} a_{ij}^{t_{ij}} \right\},
\tag{2.21}
$$

which is the capacity allocated to link $(i, j)$ to support the traffic uncertainty. $\beta_{ij}(\mathbf{a_{ij}}, \Gamma_{ij})$ can be written as the solution of the following LP:

$$
\begin{aligned}
\beta_{ij}(\mathbf{a_{ij}}, \Gamma_{ij}) = \text{maximize} \quad & \sum_{sd} k\sigma_{sd} a_{ij}^{sd} \theta_{ij}^{sd} \\
\text{subject to} \quad & \sum_{sd} \theta_{ij}^{sd} \leq \Gamma_{ij} \\
& 0 \leq \theta_{ij}^{sd} \leq 1 \qquad \forall (s,d)
\end{aligned}
\tag{2.22}
$$

The maximizing value of $\theta_{ij}$ for this LP is found by selecting the $\lfloor \Gamma_{ij} \rfloor$ largest values of $k\sigma_{sd} a_{ij}^{sd}$, and setting the corresponding $\theta_{ij}^{sd}$ variables to 1. The next largest value of $k\sigma_{sd}a_{ij}^{sd}$ is set to $\Gamma_{ij} - \lfloor \Gamma_{ij} \rfloor$. This selection corresponds directly to the selection of $S_{ij}$ and $t_{ij}$ in (2.21) respectively.

Consider the Lagrangian formed by relaxing the two sets of constraints, and adding a dual variable $z_{ij}$ for the first constraint and $p_{ij}^{sd}$ for the individual $0 \leq \theta_{ij}^{sd} \leq 1$ constraints. The resulting function is

$$\begin{aligned} L(\boldsymbol{\theta_{ij}}, z_{ij}, \mathbf{p_{ij}}) &= \sum_{sd} k\sigma_{sd}a_{ij}^{sd}\theta_{ij}^{sd} + z_{ij}(\Gamma_{ij} - \sum_{sd} \theta_{ij}^{sd}) + \sum_{sd} p_{ij}^{sd}(1 - \theta_{ij}^{sd}) \\ &= \sum_{sd}(k\sigma_{sd}a_{ij}^{sd} - z_{ij} - p_{ij}^{sd})\theta_{ij}^{sd} + z_{ij}\Gamma_{ij} + \sum_{sd} p_{ij}^{sd} \end{aligned} \tag{2.23}$$

The dual problem is $\min_{z_{ij}, \mathbf{p_{ij}} \geq 0} \max_{\boldsymbol{\theta_{ij}} \geq 0} L(\boldsymbol{\theta_{ij}}, z_{ij}, \mathbf{p_{ij}})$. Furthermore, the term $k\sigma_{sd}a_{ij}^{sd} - z_{ij} - p_{ij}^{sd}$ in (2.23) must be non-positive so the dual problem is feasible. Thus, the solution of the inner optimization is achieved when $\theta_{ij}^{sd} = 0$. Consequently, the following is the dual to (2.22).

$$\begin{aligned} \text{minimize} \quad & z_{ij}\Gamma_{ij} + \sum_{sd} p_{ij}^{sd} \\ \text{subject to} \quad & z_{ij} + p_{ij}^{sd} \geq k\sigma_{sd}a_{ij}^{sd} \quad \forall(s,d) \\ & z_{ij} \geq 0 \\ & p_{ij}^{sd} \geq 0 \quad \forall(s,d) \end{aligned} \tag{2.24}$$

By the strong duality theory, there is zero duality gap between Problems (2.22) and (2.24). Consequently, $\beta_{ij}(\mathbf{a_{ij}}, \Gamma_{ij})$ is equal to the optimal objective function value of (2.24).

Since the minimization of $C_{\max}$ in (2.20) requires the indirect minimization of $\beta_{ij}(\mathbf{a_{ij}}, \Gamma_{ij})$, (2.24) can be substituted into (2.20) as follows.

$$
\begin{array}{ll}
\text{Minimize:} & C_{\max} \\
\text{Subject To:} & C_{ij} \geq \sum_{sd} a_{ij}^{sd}\mu_{sd} + \sum_{sd} p_{ij}^{sd} + z_{ij}\Gamma_{ij} \quad \forall i,j \\
& z_{ij} + p_{ij}^{sd} \geq k\sigma_{sd}a_{ij}^{sd} \quad \forall i,j,(s,d) \\
& \sum_{j} a_{ij}^{sd} - \sum_{j} a_{ji}^{sd} = \begin{cases} 1, & \text{if } s = i \\ -1, & \text{if } d = i \\ 0, & \text{otherwise} \end{cases} \quad \forall s,d,i \\
& C_{ij} \leq C_{max} \qquad \forall (i,j) \\
& z_{ij} \geq 0 \quad \forall i,j \\
& p_{ij}^{sd} \geq 0 \quad \forall i,j,s,d \\
& a_{ij}^{sd} \geq 0 \quad \forall i,j,s,d
\end{array}
\tag{2.25}
$$

Formulation (2.25) is an LP to implement robust optimization, parameterized by $\Gamma_{ij}$. Next, we determine the values of $\Gamma_{ij}$ required to meet the constraint in (2.6).

### 2.4.2 Probability Bounds

The formulation in (2.25) solves the routing and capacity allocation subproblems for truncated normal random variables with a "small" probability of capacity constraint violation. Bertsimas provides an upper bound on that probability as a function of $\Gamma_{ij}$. We modify this bound for truncated Gaussians to relate $\epsilon$ to $\Gamma_{ij}$ for each link.

This bound is shown below through the following theorem.

**Theorem 1.** *Let $\lambda_B^{sd}$ be the traffic from source $s$ to destination $d$. Further, let $\lambda_B^{sd}$ be a continuous random variable with density $\mathcal{N}(\lambda_B^{sd}|\lambda_B^{sd} \leq \mu_{sd} + k\sigma_{sd}; \mu_{sd}, \sigma_{sd}^2)$. Let $0 \leq a_{ij}^{sd} \leq 1$ and $C_{ij} \geq 0$ satisfy*

$$
C_{ij} \geq \sum_{sd\in\mathcal{D}} \mu_{sd}a_{ij}^{sd} + \max_{S_{ij}\cup\{t_{ij}\}|S_{ij}\in\mathcal{D},|S_{ij}|=\lfloor\Gamma_{ij}\rfloor,t_{ij}\in\mathcal{D}\setminus S_{ij}} \left\{ \sum_{(s,d)\in S_{ij}} k\sigma_{sd}a_{ij}^{sd} + (\Gamma_{ij} - \lfloor\Gamma_{ij}\rfloor)k\sigma_{t_{ij}}a_{ij}^{t_{ij}} \right\}
\tag{2.26}
$$

*Let $n_{ij}$ be the number of demands routed over link $(i, j)$. Then, the probability that link $(i, j)$ overflows is bounded by*

$$\mathbf{P}\Big(\sum_{sd} \lambda_B^{sd} a_{ij}^{sd} > C_{ij}\Big) \leq \exp\Big(-\frac{\Gamma_{ij}^2 k^2}{2n_{ij}}\Big) \tag{2.27}$$

**Lemma 1.** *Let $X$ be a continuous random variable with PDF $\mathcal{N}(x|a \leq x \leq b; \mu, \sigma^2)$. Let $Y = cX + d$ for $c > 0$. $Y$ is then distributed with density $\mathcal{N}(x|ac + d \leq x \leq bc + d; c\mu + d, (c\sigma)^2)$.*

*Proof.* $X$ has PDF $f_X(x; \mu, \sigma^2, a, b) = \frac{\frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right)-\Phi\left(\frac{a-\mu}{\sigma}\right)}$, where $\phi()$ and $\Phi()$ are the PDF and CDF of a standard normal random variable respectively. Since $Y$ is a linear function of $X$, we can write the density of $Y$ as $f_Y(y) = \frac{1}{|c|} f_X(\frac{y-d}{c}; \mu, \sigma^2, a, b)$ By plugging in the definition of $f_X()$,

$$f_Y(y) = \frac{\frac{1}{c\sigma}\phi\left(\frac{y-d-c\mu}{c\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)} \tag{2.28}$$

Define $\mu' \triangleq c\mu + d$ and $\sigma' \triangleq c\sigma$

$$\begin{aligned} f_Y(y) &= \frac{\frac{1}{\sigma'}\phi\left(\frac{y-\mu'}{\sigma'}\right)}{\Phi\left(\frac{bc+d-\mu'}{\sigma'}\right) - \Phi\left(\frac{ac+d-\mu'}{\sigma'}\right)} && (2.29) \\ &= \mathcal{N}(x|ac + d \leq x \leq bc + d; c\mu + d, (c\sigma)^2) && (2.30) \end{aligned}$$

□

*Proof of Theorem 1.* Define

$$z_B^{sd} \triangleq \frac{\lambda_B^{sd} - \mu_{sd}}{\sigma_{sd}} \tag{2.31}$$

Further, define $\eta^{sd} \triangleq \frac{1}{k} z_B^{sd}$. By Lemma 1, $z_B^{sd}$ has a PDF given by $\mathcal{N}(z|z \leq k; 0, 1)$, and $\eta^{sd}$ has a PDF given by $\mathcal{N}(\eta|\eta \leq 1; 0, 1/k^2)$.

From (2.31) and the definition of $\eta^{sd}$,

$$\mathbf{P}\Big(\sum_{sd} a_{ij}^{sd}\lambda_B^{sd} > C_{ij}\Big) = \mathbf{P}\Big(\sum_{sd} a_{ij}^{sd}\mu_{sd} + \sum_{sd} a_{ij}^{sd}\sigma_{sd}z_B^{sd} > C_{ij}\Big) \tag{2.32}$$

$$= \mathbf{P}\Big(\sum_{sd} a_{ij}^{sd}\mu_{sd} + \sum_{sd} a_{ij}^{sd}\sigma_{sd}k\eta^{sd} > C_{ij}\Big) \tag{2.33}$$

Since $C_{ij}$ satisfies (2.26),

$$\mathbf{P}\Big(\sum_{sd} a_{ij}^{sd}\lambda_B^{sd} > C_{ij}\Big) \leq \mathbf{P}\Big(\sum_{sd} a_{ij}^{sd}k\sigma_{sd}\eta^{sd} > \sum_{(s,d)\in S_{ij}} a_{ij}^{sd}k\sigma_{sd} + (\Gamma_{ij} - \lfloor\Gamma_{ij}\rfloor)a_{ij}^{t_{ij}}k\sigma_{t_{ij}}\Big) \tag{2.34}$$

$$= \mathbf{P}\Big(\sum_{sd} a_{ij}^{sd}\sigma_{sd}\eta^{sd} > \sum_{(s,d)\in S_{ij}} a_{ij}^{sd}\sigma_{sd} + (\Gamma_{ij} - \lfloor\Gamma_{ij}\rfloor)a_{ij}^{t_{ij}}\sigma_{t_{ij}}\Big) \tag{2.35}$$

Moving terms with demands $(s,d) \in S_{ij}$ to the right of the inequality,

$$= \mathbf{P}\Big(\sum_{(s,d)\notin S_{ij}} a_{ij}^{sd}\sigma_{sd}\eta^{sd} > \sum_{(s,d)\in S_{ij}} a_{ij}^{sd}\sigma_{sd}(1-\eta^{sd}) + (\Gamma_{ij} - \lfloor\Gamma_{ij}\rfloor)a_{ij}^{t_{ij}}\sigma_{t_{ij}}\Big) \tag{2.36}$$

Let $r = \operatorname{argmin}_{r'\in S_{ij}\bigcup t_{ij}} a_{ij}^{r'}\sigma_{r'}$. Since $\eta^{sd} \leq 1$,

$$\leq \mathbf{P}\Big(\sum_{(s,d)\notin S_{ij}} a_{ij}^{sd}\sigma_{sd}\eta^{sd} > a_{ij}^{r}\sigma_r\Big(\sum_{(s,d)\in S_{ij}} (1-\eta^{sd}) + \Big(\Gamma_{ij} - \lfloor\Gamma_{ij}\rfloor\Big)\Big)\Big) \tag{2.37}$$

$$= \mathbf{P}\Big(\sum_{(s,d)\notin S_{ij}} a_{ij}^{sd}\sigma_{sd}\eta^{sd} + \sum_{(s,d)\in S_{ij}} a_{ij}^{r}\sigma_r\eta^{sd} > a_{ij}^{r}\sigma_r\Gamma_{ij}\Big) \tag{2.38}$$

$$= \mathbf{P}\Big(\sum_{(s,d)} \gamma_{ij}^{sd}\eta^{sd} > \Gamma_{ij}\Big) \tag{2.39}$$

where

$$\gamma_{ij}^{sd} = \begin{cases} 1, & \text{if } (s,d) \in S_{ij}\bigcup t_{ij} \\ \frac{a_{ij}^{sd}\sigma_{sd}}{a_{ij}^{r}\sigma_r}, & \text{if } (s,d) \notin S_{ij}\bigcup t_{ij} \end{cases} \tag{2.40}$$

Since $r$ is chosen to be an element of $S_{ij} \bigcup t_{ij}$, then $a_{ij}^{r}\sigma_r \geq a_{ij}^{sd}\sigma_{sd} \quad \forall(s,d) \notin S_{ij}$. Therefore, $\gamma_{ij}^{sd}$ defined in (2.40) is always less than or equal to 1. An additional Lemma is needed to complete the proof.

**Lemma 2.**

$$\mathbf{E}[e^{t\eta^{sd}}] = M^{sd}(t) = \exp\left(\frac{t^2}{2k^2}\right)\left(\frac{\Phi(k - t/k)}{\Phi(k)}\right). \tag{2.41}$$

*Proof.* Let $Y \sim \mathcal{N}(0, \frac{1}{k^2})$,

$$M^{sd}(t) = \mathbf{E}[e^{tY} | Y \leq 1] \tag{2.42}$$

$$= \int_{-\infty}^{1} \frac{e^{ty} f_Y(y)dy}{\Phi(k)} \tag{2.43}$$

$$= \frac{1}{\Phi(k)} \int_{-\infty}^{1} e^{ty} \frac{k}{\sqrt{2\pi}} e^{-\frac{y^2k^2}{2}} dy \tag{2.44}$$

$$= \frac{k}{\Phi(k)\sqrt{2\pi}} \int_{-\infty}^{1} \exp\left( - \frac{k^2}{2}(y^2 - \frac{2ty}{k^2})\right) dy \tag{2.45}$$

$$= \frac{k}{\Phi(k)\sqrt{2\pi}} \int_{-\infty}^{1} \exp\left( - \frac{k^2}{2}\left(y^2 - \frac{2ty}{k^2} + \frac{t^2}{k^4}\right) + \frac{t^2}{2k^2}\right) dy \tag{2.46}$$

$$= \frac{k}{\Phi(k)\sqrt{2\pi}} \int_{-\infty}^{1} \exp\left(\frac{t^2}{2k^2}\right) \exp\left( - \frac{k^2}{2}\left(y - \frac{t}{k^2}\right)^2\right) dy \tag{2.47}$$

$$= \frac{e^{\frac{t^2}{2k^2}}}{\Phi(k)} \int_{-\infty}^{1} \frac{k}{\sqrt{2\pi}} e^{-\frac{k^2}{2}(y - \frac{t}{k^2})^2} dy \tag{2.48}$$

This can be reduced in terms of the standard normal CDF.

$$= \frac{e^{\frac{t^2}{2k^2}}}{\Phi(k)} \left(\Phi\left(\frac{1 - \frac{t}{k^2}}{\frac{1}{k}}\right) - \Phi\left(\frac{-\infty - \frac{t}{k^2}}{\frac{1}{k}}\right)\right) \tag{2.49}$$

$$= e^{\frac{t^2}{2k^2}} \left(\frac{\Phi(k - \frac{t}{k})}{\Phi(k)}\right) \tag{2.50}$$

□

This lemma can be used to complete the proof. Starting with the bound in (2.39),

$$\mathbf{P}\left(\sum_{(s,d)} \gamma_{ij}^{sd}\eta^{sd} > \Gamma_{ij}\right)$$

$$= \mathbf{P}\left(t\sum_{(s,d)} \gamma_{ij}^{sd}\eta^{sd} > t\Gamma_{ij}\right). \qquad \forall t \geq 0 \tag{2.51}$$

$$= \mathbf{P}\left(\exp\left(t\sum_{(s,d)} \gamma_{ij}^{sd}\eta^{sd}\right) > \exp\left(t\Gamma_{ij}\right)\right) \tag{2.52}$$

$$\leq \frac{\mathbf{E}[e^{t\sum_{(s,d)}\gamma_{ij}^{sd}\eta^{sd}}]}{e^{t\Gamma_{ij}}} \qquad \text{By Markov's Inequality.} \tag{2.53}$$

$$= \prod_{sd} \frac{\mathbf{E}[e^{t\eta^{sd}\gamma_{ij}^{sd}}]}{e^{t\Gamma_{ij}}} \qquad \text{By independence.} \tag{2.54}$$

$$= \prod_{sd} \frac{M^{sd}(t\gamma_{ij}^{sd})}{e^{t\Gamma_{ij}}} \tag{2.55}$$

$$= \prod_{sd} \frac{e^{\frac{(t\gamma_{ij}^{sd})^2}{2k^2}}\left(\frac{\Phi(k-\frac{t\gamma_{ij}^{sd}}{k})}{\Phi(k)}\right)}{e^{t\Gamma_{ij}}} \qquad \text{By Lemma 2} \tag{2.56}$$

$$\leq \prod_{sd} \frac{e^{\frac{(t\gamma_{ij}^{sd})^2}{2k^2}}}{e^{t\Gamma_{ij}}} \tag{2.57}$$

Since $\Phi(k) \geq \Phi(k - \frac{t\gamma_{ij}^{sd}}{k}) \quad \forall t \geq 0, \gamma_{ij}^{sd} \geq 0.$

$$= \frac{\exp\left(\frac{t^2}{2k^2}\sum_{sd}(\gamma_{ij}^{sd})^2\right)}{e^{t\Gamma_{ij}}} \tag{2.58}$$

$$\leq \exp\left(\frac{n_{ij}t^2}{2k^2} - t\Gamma_{ij}\right) \qquad \text{Since } 0 \leq \gamma_{ij}^{sd} \leq 1. \tag{2.59}$$

Find $t$ to maximize exponent:

$$\frac{d}{dt}\left(\frac{n_{ij}t^2}{2k^2} - t\Gamma_{ij}\right) = \frac{n_{ij}t}{k^2} - \Gamma_{ij} = 0 \tag{2.60}$$

$$t = \frac{\Gamma_{ij}k^2}{n_{ij}} \tag{2.61}$$

Plug in $t$ into bound:

$$\mathbf{P}\Big(\sum_{(s,d)} \gamma_{ij}^{sd}\eta^{sd} > \Gamma_{ij}\Big) \leq \exp\left(\frac{n_{ij}}{2k^2}\frac{\Gamma_{ij}^2 k^4}{n_{ij}^2} - \frac{\Gamma_{ij}k^2}{n_{ij}}\Gamma_{ij}\right) \tag{2.62}$$

$$= \exp\left( - \frac{\Gamma_{ij}^2 k^2}{2n_{ij}}\right) \tag{2.63}$$

□

The variable $n_{ij}$ is upper bounded by the number of demands in the network, but specifying $n_{ij}$ separately for each link results in a tighter bound. This is addressed in Section 2.4.4.

Theorem 1 yields a bound on the probability of constraint violation that is independent of the traffic distribution. Given a desired probability of overflow on a link $\epsilon$, the value of the parameter $\Gamma_{ij}$ for that link can be computed by

$$\Gamma_{ij} = \frac{\sqrt{-2n_{ij}\log\epsilon}}{k}. \tag{2.64}$$

Since capacity is assigned to a link as a linear function of $\Gamma_{ij}$, capacity grows with $\sqrt{n_{ij}}$. This is similar to the optimal convex capacity constraint from (2.8). Therefore, statistical multiplexing of traffic is beneficial for reducing capacity provisioning in this formulation.

Figure 2-6 plots the bound in Theorem 1 as a function of $\Gamma_{ij}$ for $n_{ij} = 20$ and different values of $k$. For example, if $k = 2$, allocating capacity to allow for an $\epsilon = 0.05$ probability of error requires $\Gamma_{ij} = 5.75$. By accepting this small probability of constraint violation, the capacity required is reduced by almost a factor of 4 as compared to the approach in Section 2.3. As the value of $k$ increases, the traffic distributions are truncated further from the mean, and smaller values of $\Gamma_{ij}$ are required.

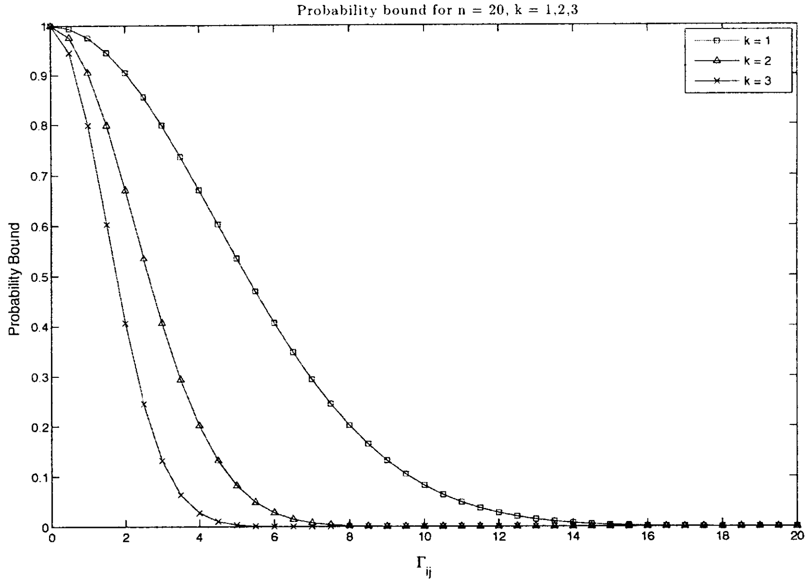


Figure 2-6: Upper bound on probability of link overflow as a function of the parameter $\Gamma_{ij}$

### 2.4.3 Robust Approach for Gaussian Demands

In Sections 2.4.1 and 2.4.2, a robust optimization formulation and associated probability bounds on constraint violation were developed specific to truncated Gaussian demands. In this section, we extend this to unbounded Gaussian demands. The LP presented in (2.25) requires no modification; however, the probability bound in Theorem 1 depends on the distribution of the random demands.

Recall that the demands $\lambda_B^{sd}$ are upper bounded at $\mu_{sd} + k\sigma_{sd}$. In the capacity constraint in (2.20), $\Gamma_{ij}$ corresponds to the number of demands that need to be protected, and a capacity of $k\sigma_{sd}a_{ij}^{sd}$ must be allocated to account for the worst case realization for each of those demands. In equation (2.64), $k$ and $\Gamma_{ij}$ are inversely proportional, implying that as $k$ increases, fewer demands need protection.

If $\{X_k, k \geq 0\}$ is an independent sequence of continuous random variables, with $X_k$ distributed according to a truncated gaussian described by (2.19), then the sequence of $X_k$ converges in distribution to a normal random variable with mean $\mu$ and variance $\sigma^2$.

The proof of Theorem 1 holds for all finite values of $k$. As $k$ becomes large, $\Gamma_{ij}$ eventually satisfies $0 \leq \Gamma_{ij} < 1$. This follows directly from equation (2.64), as $\epsilon$ is fixed and $n_{ij}$ is upper bounded by the total number of demands. Since $\Gamma_{ij}$ is less than one, $S_{ij}$ in (2.21) is the empty set, and equation (2.21) can be rewritten as

$$\beta_{ij}(\mathbf{a_{ij}}, \Gamma_{ij}) = \Gamma_{ij} k \max_{sd}\{\sigma_{sd} a_{ij}^{sd}\}. \tag{2.65}$$

The maximization in equation (2.21) is now over one $s$-$d$ pair representing $t_{ij}$. Combining (2.64) and (2.65) yields a new equation for $\beta_{ij}$, independent of both $k$ and $\Gamma_{ij}$.

$$\beta_{ij}(\mathbf{a_{ij}}) = \max_{sd}\{\sigma_{sd} a_{ij}^{sd}\}\sqrt{-2n_{ij}\log\epsilon} \tag{2.66}$$

Formulation (2.20) is rewritten as the following LP.

$$\begin{aligned}
\text{Minimize:}\quad & C_{\max} \\
\text{Subject To:}\quad & C_{ij} \geq \sum_{sd\in\mathcal{D}} \mu_{sd} a_{ij}^{sd} + z_{ij}\sqrt{-2n_{ij}\log\epsilon} \qquad \forall(i,j) \\
& \sum_j a_{ij}^{sd} - \sum_j a_{ji}^{sd} = \begin{cases} 1, & \text{if } s = i \\ -1, & \text{if } d = i \\ 0, & \text{otherwise} \end{cases} \qquad \forall(s,d), i \\
& z_{ij} \geq \sigma_{sd} a_{ij}^{sd} \qquad \forall(s,d) \\
& C_{ij} \leq C_{max} \qquad \forall(i,j) \\
& a_{ij}^{sd} \geq 0 \quad \forall i,j,s,d
\end{aligned} \tag{2.67}$$

the variable $z_{ij}$ has been introduced to represent the maximum value of $a_{ij}^{sd}\sigma_{sd}$. Also, the $n_{ij}$ must be optimization parameters in order for (2.67) to be an LP.

The solution to this formulation differs from the solutions to (2.9) and (2.12). Link capacity in (2.67) is a function of the largest variance of the demands, rather than the variances of all the demands. Let the vectors $\mu$ and $\sigma$ be the mean and standard deviation of the traffic, where an element of each vector corresponds to a specific demand. Additionally, let $\mathbf{a_{ij}}$ be the vector with elements $a_{ij}^{sd}$. Then, the capacity constraint (2.8) can be rewritten as

$$C_{ij} \geq \|\langle \mathbf{a_{i,j}}, \mu \rangle\|_1 + k\|\langle \mathbf{a_{i,j}}, \sigma \rangle\|_2. \tag{2.68}$$

In equation (2.68), $k = \Phi^{-1}(1-\epsilon)$ and $\langle \cdot \rangle$ represents a vector inner product. Additionally, $\|\cdot\|_p$ is the $p$-norm satisfying $\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$. The capacity constraint in (2.66) can be expressed in the same way.

$$C_{ij} \geq \|\langle \mathbf{a_{i,j}}, \mu \rangle\|_1 + c\sqrt{n_{ij}}\|\langle \mathbf{a_{i,j}}, \sigma \rangle\|_\infty. \tag{2.69}$$

The constant $c$ equals $\sqrt{2\log\epsilon}$ and $\|\cdot\|_\infty$ is the infinity (maximum) norm. For the purpose of comparison, ignore the constants $k$ and $c$, and the first term in each equation. The optimal formulation chooses a routing vector $\mathbf{a_{ij}}$ with a minimal 2-norm, while the new robust formulation minimizes $\sqrt{n_{ij}}\|\langle \mathbf{a_{i,j}}, \sigma \rangle\|_\infty$.

To clarify, consider the following hypothetical scenario. Assume there are six i.i.d flows (A through F) that can be routed over a link. Method 1 is to route all of flow A, and half of flow B over the link under analysis. Method 2 is to route all of flow A, and 10% of flows B through F over that link. If all the flows have the same distribution $\lambda \sim \mathcal{N}(\mu, \sigma^2)$, then according to (2.8), a capacity of $1.5\mu + 1.12k\sigma$ is required on that link for the first method, and $1.5\mu + 1.025k\sigma$ is required for the second. However, according to (2.67), a capacity of $1.5\mu + \sqrt{2}c\sigma$ is required for the first method, and $1.5\mu + \sqrt{5}c\sigma$ is required for the second. In this example, the optimal routing decisions are different between the two formulations and the optimal.

### 2.4.4 Implementation

Subtleties in the formulation in (2.67) prevent a straightforward implementation. The parameters $n_{ij}$ must be specified a priori; however, these parameters depend on the optimal values of the variables $a_{ij}^{sd}$. The solution remains feasible if the $n_{ij}$ are replaced with an upper bound (i.e. the total number of demands in the network), but a loose bound results in capacity over-provisioning. A better solution is obtainable by restricting the paths on which each demand can be routed through the network, as it allows for a tighter upper bound on $n_{ij}$.

### MILP Formulation for Optimal Routing

Integer constraints can be used to compute $n_{ij}$ directly and avoid the need for extra path restrictions. Let $f_{ij}^{sd}$ be a binary variable satisfying $f_{ij}^{sd} \geq a_{ij}^{sd}$. Thus, $f_{ij}^{sd}$ equals 1 if $a_{ij}^{sd} > 0$, and 0 otherwise. Let $x_{ij}^{m} = 1$ if there are $m$ demands traversing link $(i, j)$, and 0 otherwise. These variables satisfy

$$\sum_{sd} f_{ij}^{sd} = \sum_{m=0}^{D} m x_{ij}^{m} \qquad \forall (i,j) \tag{2.70}$$

where D is the total number of demands in the network. By combining this constraint with one forcing only one of $\{x_{ij}^{0}, x_{ij}^{1}, \ldots, x_{ij}^{D}\}$ to be equal to one for each link $(i, j)$, the variables $x_{ij}^{m}$ specify the number of flows on each link. Therefore, the capacity constraint of (2.67) can be rewritten as

$$C_{ij} \geq \sum_{sd \in \mathcal{D}} \mu_{sd} a_{ij}^{sd} + z_{ij} \sqrt{-2 \log \epsilon} \sum_{m=0}^{D} x_{ij}^{m} \sqrt{m} \qquad \forall (i,j). \tag{2.71}$$

Equation (2.71) is nonlinear, since $z_{ij}$ and $x_{ij}^{m}$ are both optimization variables. However, (2.71) can be linearized by introducing a variable $y_{ij}^{m}$ to represent the product $z_{ij} x_{ij}^{m}$ with the following constraints.

$$y_{ij}^{m} \geq z_{ij} + M(x_{ij}^{m} - 1) \qquad \forall (i,j), m \tag{2.72}$$

$$y_{ij}^{m} \leq M x_{ij}^{m} \qquad \forall (i,j), m \tag{2.73}$$

$$y_{ij}^{m} \geq 0 \qquad \forall (i,j), m \tag{2.74}$$

In the above equations, $M$ is a large number such that $M > \max_{sd} \sigma_{sd}$. When $x_{ij}^{m} = 0$, then $x_{ij}^{m} z_{ij} = 0$, and constraints (2.73) and (2.74) force $y_{ij}^{m}$ to be 0. On the other hand, if $x_{ij}^{m} = 1$, constraint (2.72) will force $y_{ij}^{m} \geq z_{ij}$, which at the optimal

$$
\begin{aligned}
\text{Minimize:}\quad & C_{\max} \\
\text{Subject To:}\quad & C_{ij} \geq \sum_{sd\in\mathcal{D}} \mu_{sd}a_{ij}^{sd} + \sqrt{-2\log\epsilon}\sum_{m=0}^{D}\sqrt{m}y_{ij}^{m} \qquad \forall(i,j) \\
& \sum_j a_{ij}^{sd} - \sum_j a_{ji}^{sd} = \begin{cases} 1, & \text{if } s=i \\ -1, & \text{if } d=i \\ 0, & \text{otherwise} \end{cases} \qquad \forall(s,d),i \\
& C_{ij} \leq C_{max} \qquad \forall(i,j) \\
& f_{ij}^{sd} \geq a_{ij}^{sd} \qquad \forall(i,j),(s,d) \\
& \sum_{sd} f_{ij}^{sd} = \sum_{m=0}^{D} m x_{ij}^{m} \qquad \forall(i,j) \\
& \sum_{m=0}^{D} x_{ij}^{m} = 1 \qquad \forall(i,j) \\
& y_{ij}^{m} \geq \sigma_{sd}a_{ij}^{sd} + M(x_{ij}^{m}-1) \qquad \forall(s,d),(i,j),m \\
& y_{ij}^{m} \leq M x_{ij}^{m} \qquad \forall(i,j),m \\
& y_{ij}^{m} \geq 0 \qquad \forall(i,j),m \\
& a_{ij}^{sd} \geq 0 \quad \forall i,j,s,d \\
& f_{ij}^{sd}, x_{ij}^{m} \in \{0,1\} \qquad \forall(s,d),(i,j),m
\end{aligned}
\tag{2.75}
$$

solution will be satisfied with equality. The complete MILP to solve the routing problem is formulated in Problem (2.75).

While this MILP computes the exact values of $n_{ij}$ through the variables $x_{ij}^{m}$, the addition of the integer variables makes this program computationally intractable. Therefore, one can investigate sub-optimal design heuristics to solve (2.75) for larger networks.

The proposed heuristics are based on estimating the values $n_{ij}$, evaluating that guess, and then revising it iteratively. The optimal routing according to (2.67) changes based on the estimate of $n_{ij}$, as routing traffic over links with a high $n_{ij}$ requires more capacity, so less traffic will be routed on these links. Therefore, $n_{ij}$ is considered a link cost.

### Increasing Cost Algorithm

The first algorithm is the Increasing Cost Algorithm (ICA). All the link costs initialize to 1, the optimization problem in (2.67) is solved, and the number of demands traversing each link is counted. If there are more demands routed on a link than the cost of that link, the cost is increased by one. Then, the algorithm repeats with the new link costs. ICA terminates when each link has a higher cost than the number of demands routed on the link, at which point link capacities are computed by directly applying (2.8). This iterative scheme is guaranteed to converge in at most $N^2$ iterations, where $N$ is the number of links.

The cost on any link can increase during each iteration, but can never decrease. In the early iterations, a link may be labeled as having a high cost, thus preventing ICA from optimally routing traffic. However, each iteration only requires an LP to be solved, implying that the entire algorithm can be run in polynomial time. This is a significant improvement over the MILP in (2.75) and even the convex optimization formulation in (2.9). These timing results are shown in Figure 2-7.

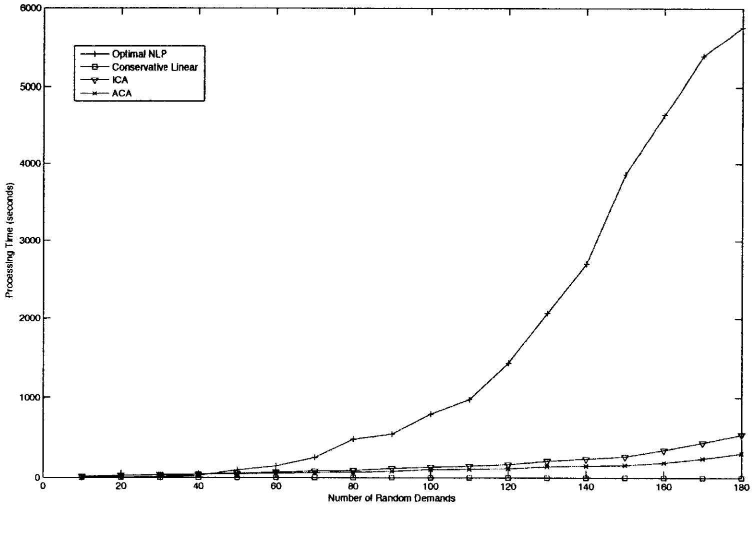


Figure 2-7: Comparison of processing time of different schemes for the NSFNET in Figure 2-4, as a function of the number of random demands. The MILP timing has been omitted as it is several orders of magnitude greater than the scale of this plot.

### Adaptive Cost Algorithm

We propose a second heuristic, the Adaptive Cost Algorithm (ACA), in which link costs can be decreased if they have been set too high. At each iteration $k$, the cost is updated according to $n_{ij}(k+1) = n_{ij}(k) - \gamma(k)(n_{ij}(k) - \sum_{sd} f_{ij}^{sd})$, where $f_{ij}^{sd}$ equals 1 if and only if link $(i,j)$ carries demand pair $(s,d)$. Also, $\gamma(k)$ is a step size as a function of the current iteration. The step size monotonically decreases to zero with $k$ slowly so that the algorithm arrives at a near-optimal solution. In this work, we use $\gamma(k) = \frac{1}{\sqrt{k}}$.

ACA terminates when the mean squared error between the old cost and the new cost falls below a positive constant. ACA is guaranteed to converge since $\lim_{k\to\infty} \gamma(k) = 0$. In order to ensure that the best solution is returned, we calculate the value of $C_{\max}$ at each iteration based on the current routing and equation (2.8), and return the best routing upon convergence.

## 2.4.5 Simulation Results

The ACA and ICA heuristics are compared to the optimal formulation in (2.9) and the overly conservative formulation in (2.12), shown in Figure 2-8 for routing traffic and allocating capacity over the NSFNET. In this simulation, demands are distributed as $\mathcal{N}(100, 35^2)$ and exist between randomly chosen node pairs. Both the ACA and ICA algorithms achieve approximately the same maximum link capacity for each random number of demands, and yield a savings of approximately 15% over the conservative formulation. However, neither heuristic performs as well as the optimal formulation, which yields 25% to 30% savings. Despite their sub-optimalities, the robust optimization heuristics offer substantially better performance than the conservative approximation.

The routing decisions made by solving each formulation differ due to the capacity constraints. Next, we analyze what fraction of the difference between the maximum link capacity results from the routing choices between the formulations. Consider modifying each of the previous network routing and capacity allocation strategies to

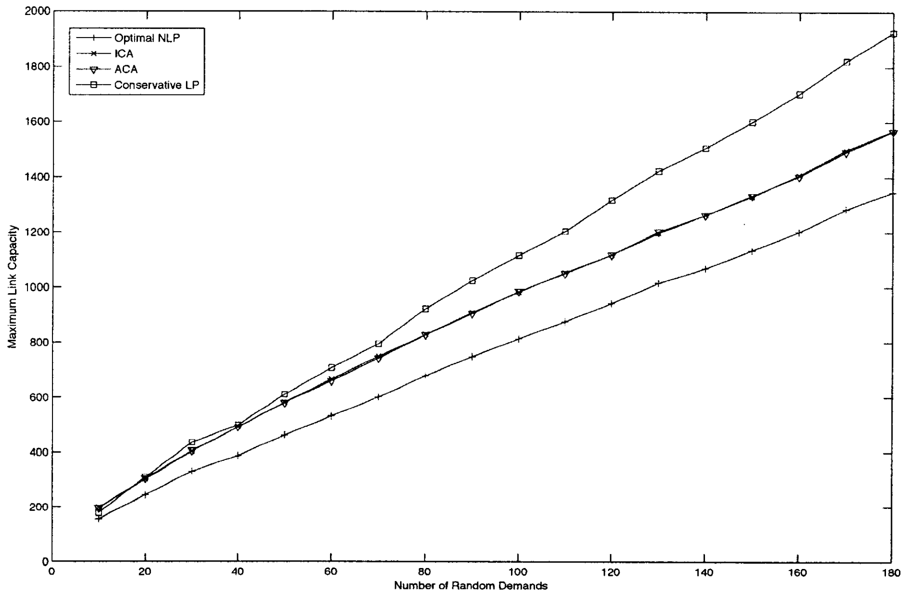


Figure 2-8: Comparison of different routing algorithms applied to the NSFNET in 2-4. Demands are normally distributed with $\mu = 100$ and $\sigma = 35$, and are between randomly generated node pairs. the number of these random demands is shown on the x-axis. Results are averaged over 5 trials.

recompute the capacity on each link using the constraint in (2.8) after the routed has been computed, preventing unnecessary over-provisioning. These modified approaches are compared to the optimal routing solution for NSFNET in Figure 2-9. The optimal non-linear formulation only provides a savings of 5% over the modified LP approach. Additionally, the modified LP approach outperforms the modified ICA algorithm by 5%.

An explanation for the poor performance of the modified ICA algorithm compared to the surprisingly good performance of the modified LP approach is that the NSFNET does not support enough opportunities for traffic from different sources to share link resources. To test this claim, we consider another network shown in Figure 2-10. We run the same simulation, with we randomly generated, normally distributed demands with $\mu = 100$, $\sigma = 35$, and compare the results of our different approaches.

Figure 2-11a shows the results of comparing the original ICA approach with the conservative linear formulation in (2.12) and the optimal solution in (2.9). The maxi-

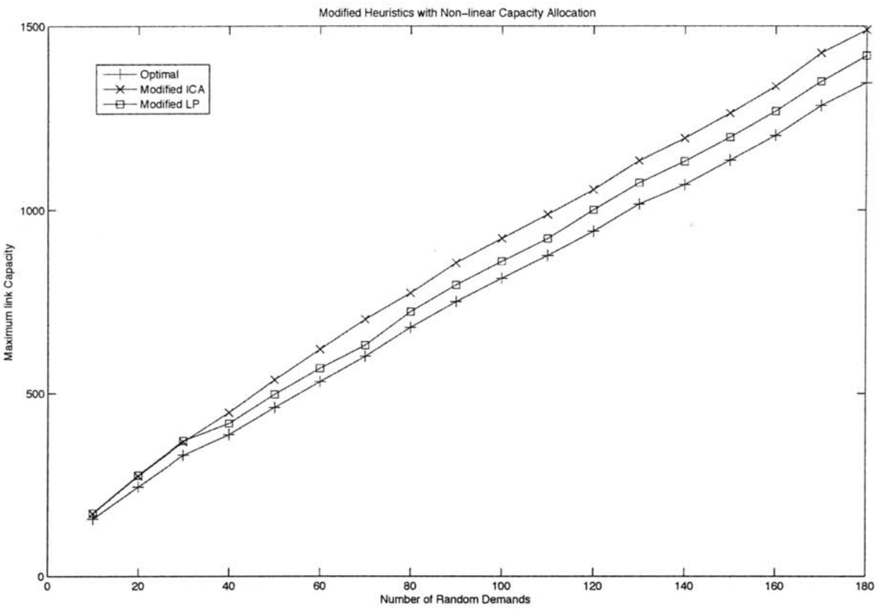


Figure 2-9: Comparison of the modified approaches to the optimal approach for NSFNET. Each algorithm is given the same set of random demands, each normally distributed with $\mu = 100$ and $\sigma = 35$.

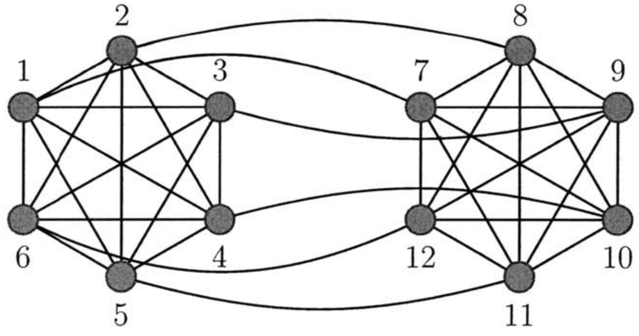


Figure 2-10: Example network providing ample opportunities for splitting traffic over multiple paths. Links in this network are bidirectional.

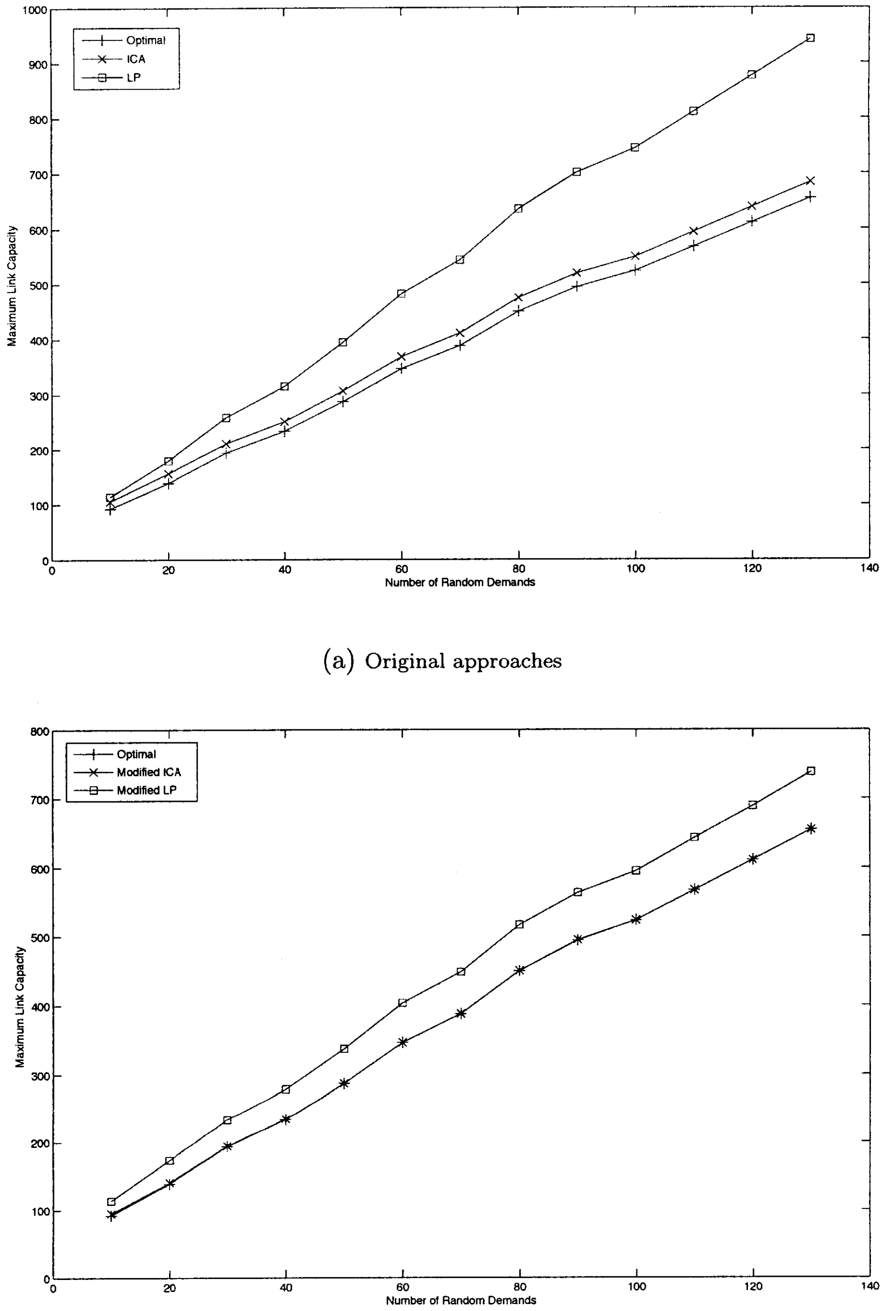
mum link capacity for the optimal solution offers a 30% savings over that required by the linear approach. In this example, the ICA heuristic performs substantially better than the LP approach, yielding a capacity savings of 25% for the most congested link. We also consider the modified LP and modified ICA approaches described above. The results of comparing these to the nonlinear optimal solution are shown in 2-11b. In this example, the modified ICA approach performs equally as well as the optimal approach. Both these approaches achieve a max-link capacity savings of 10% to 15% over the linear approach with the optimal capacity allocation modification. This gain is a result of the resource sharing opportunities made available by the network, which were unavailable when routing over NSFNET.

## 2.5 Network Design

In Section 2.2, the routing and capacity allocation problems were formulated to support a traffic matrix with Gaussian demands. Due to the non-linearity of that formulation, the link placement problem has been omitted, as it requires integer variables. Integer and linear formulations were proposed in Section 2.3 and 2.4. The remaining task is to understand how the design problem changes when accounting for stochastic traffic, and to extend the robust optimization formulations to include the link placement sub-problem.

### 2.5.1 Network Design Theory for Stochastic Traffic

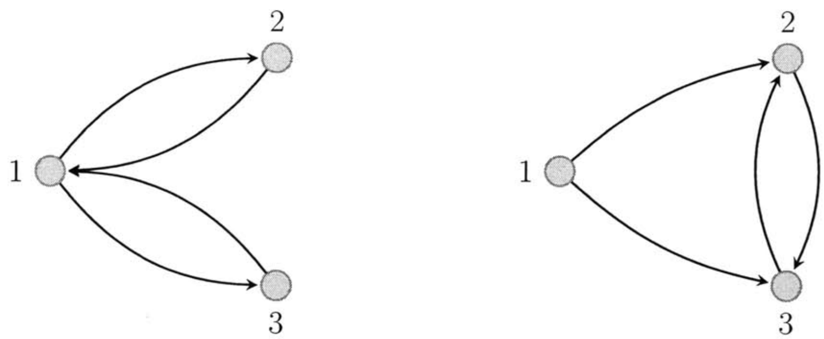
Consider a network made up of three nodes. Suppose we have i.i.d. traffic ($\mathcal{N}(10, \sigma)$) from node 1 to node 2 and node 1 to node 3. We want to design a network and route the traffic over such a network such that max link capacity is minimized, and the link placement is restricted to a limit of four links, assuming no parallel links. If traffic is deterministic ($\sigma = 0$), the optimal link placement only requires a link from 1 to 2 and 1 to 3 while the other two links can be placed arbitrarily. Thus, an optimal network topology for this example is shown in Figure 2-12a. With this link placement, the traffic can be sent on the one hop path to its destination. The required capacity is



(a) Original approaches

(b) Modified approaches

Figure 2-11: Solutions to the routing and capacity allocation problems for the network in 2-10. Demands are all distributed as $\mathcal{N}(100, 35^2)$ and are between randomly chosen node pairs. The number of these demands are shown on the x-axis. Each graph is averaged over 5 simulations.



(a) Optimal Deterministic Topology

(b) Optimal Stochastic Topology

Figure 2-12: Possible designs for a three-node network with four links, when there are two i.i.d demands from nodes 1 to 2 and 1 to 3

$C_{\max} = 10$.

Now consider the stochastic traffic case, by letting $\sigma = 1$. The optimal topology is shown in Figure 2-12b. On this topology, half of each demand can be sent on each link, and a lower $C_{\max}$ is achievable than on the network in Figure 2-12a. To be precise, the solution to (2.9) is $C_{\max} = 10 + \frac{\Phi^{-1}(1-\epsilon)}{\sqrt{2}}$. If the first topology is used to route the stochastic traffic instead, it would be sent directly to the destination, and $C_{\max} = 10 + \Phi^{-1}(1-\epsilon)$.

Consider the effect of routing deterministic traffic over the optimal topology for stochastic traffic in Figure 2-12b. If the traffic is split over both paths to each destination, the required capacity is $C_{\max} = 10$, the same as if the traffic was directly sent over the single hop paths. As this example suggests, there are possibly many topologies to optimally route deterministic traffic, but fewer optimal topologies for stochastic traffic. Furthermore, in this example the optimal topology for stochastic demands is also an optimal topology for deterministic demands.

We have shown, that an optimal topology for deterministic traffic may not be optimal for stochastic traffic. However, the inverse is true in this example. Further, as the variance of the traffic in the above example increases, the optimal topology remains the same. These observations are generalized in the following conjecture.

**Conjecture 1.** *Given a set of nodes and a traffic matrix where the demand between each node pair is normally distributed with mean $\mu_{sd}$ and standard deviation $\sigma_{sd} > 0$,* *there exists a topology $T^*$ that is optimal for all values of $\sigma$.*

Conjecture 1 suggests that an optimal topology for stochastic traffic of some positive variance will be optimal for traffic of any variance, including deterministic traffic. However, there may exist other optimal topologies for deterministic traffic as well. Hence, of the topologies that are optimal for deterministic traffic, a subset of them will be optimal for Gaussian traffic, of any variance. This remark is summarized below.

*Remark.* Let $\mathcal{T}$ be the set of optimal topologies for deterministic traffic and let $T_\sigma$ be the set of optimal topologies for stochastic traffic with standard deviation $\sigma > 0$. Then, $T_\sigma$ is the same for all $\sigma > 0$, and $T_\sigma \subseteq \mathcal{T}$

This claim will not be proven, but there is strong intuitive evidence to suggest its validity. Recall from Section 2.3.1 that for stochastic traffic, there is an increased interest in combining small fractions of traffic from different sources. We refer to this as *resource-sharing.* The amount of traffic combined on each link varies with $\sigma$, but the number of demands routed over a link remains constant as $\sigma$ changes. Note that if the traffic is deterministic, then resource-sharing is not beneficial. A topology that is optimal for traffic demands with positive variance will place links in such a way that the amount of resource sharing available is maximized. A weaker version of Conjecture 1 is proven through the following theorem.

**Theorem 2.** *Given a set of nodes and a traffic matrix where the demand between each node pair is normally distributed with mean $\mu_{sd}$ and standard deviation $\sigma$, $\exists \sigma_0 > 0$ such that for any $\sigma$ satisfying $0 \leq \sigma \leq \sigma_0$, the optimal topology designed for that $\sigma$ is also optimal for deterministic traffic.*

*Proof.* Let $T_0$ and $T_\sigma$ be the optimal topologies for deterministic traffic and Gaussian traffic with variance $\sigma^2$ respectively. Further, let $C_{\max}^{T,\sigma}$ be the optimal maximum link capacity in routing traffic with demand variance $\sigma^2$ over topology $T$. For example, $C_{\max}^{T_0,0}$ is the solution for routing deterministic traffic over the topology that is optimal for deterministic traffic. Throughout this proof, we will use $\kappa = \Phi^{-1}(1-\epsilon)$, given a design parameter $\epsilon$. We will prove our claim by contradiction.

Suppose $T_\sigma$ is worse than $T_0$ in terms of routing deterministic traffic. That is,

$$C_{\max}^{T_0,0} < C_{\max}^{T_\sigma,0}. \tag{2.76}$$

The difference between these two solutions $\Delta$ is defined as

$$\Delta = C_{\max}^{T_\sigma,0} - C_{\max}^{T_0,0} > 0. \tag{2.77}$$

Let $a_{ij}^{sd}$ and $b_{ij}^{sd}$ be the optimal routing for traffic with variance $\sigma^2$ and deterministic traffic on topology $T_0$ respectively. Then by the capacity constraint in (2.8),

$$C_{\max}^{T_0,\sigma} = \sum_{sd} a_{ij}^{sd}\mu_{sd} + \kappa\sigma\sqrt{\sum_{sd}(a_{ij}^{sd})^2} \leq \sum_{sd} b_{kl}^{sd}\mu_{sd} + \kappa\sigma\sqrt{\sum_{sd}(b_{kl}^{sd})^2} \tag{2.78}$$

where $(i,j)$ is a link achieving max capacity when stochastic traffic is optimally routed over $T_0$, and $(k,l)$ is a link achieving max capacity when deterministic traffic is optimally routed over $T_0$. Therefore,

$$C_{\max}^{T_0,\sigma} \leq C_{\max}^{T_0,0} + \kappa\sigma\sqrt{\sum_{sd}(b_{kl}^{sd})^2} \tag{2.79}$$

$$\leq C_{\max}^{T_0,0} + \kappa\sigma\sqrt{D} \tag{2.80}$$

where $D$ is a constant representing the maximum value of $\sum_{sd}(b_{kl}^{sd})^2$ for any feasible routing.

Consider the following optimization problem for routing deterministic traffic over any fixed topology $T$.

$$\begin{aligned}
\text{Minimize:} \quad & C_{\max} \\
\text{Subject To:} \quad & C_{\max} \geq \sum_{sd} a_{ij}^{sd} \mu_{sd} \qquad \forall (i,j) \in \mathcal{L} \\
& \sum_j a_{ij}^{sd} - \sum_j a_{ji}^{sd} = \begin{cases} 1, & \text{if } s = i \\ -1, & \text{if } d = i \\ 0, & \text{otherwise} \end{cases} \qquad \forall s, d, i \\
& a_{ij}^{sd} \geq 0 \qquad \forall (s,d), (i,j)
\end{aligned} \tag{2.81}$$

The Lagrangian of problem (2.81) is

$$\begin{aligned}
L(\nu, \theta, C_{\max}, a) = C_{\max} &+ \sum_{ij} \nu_{ij} (\sum_{sd} \mu_{sd} a_{ij}^{sd} - C_{\max}) \\
&+ \sum_{s,d,i} \theta_i^{sd} (\beta_i^{sd} - h_i^{sd}(\mathbf{a^{sd}}))
\end{aligned} \tag{2.82}$$

where $\nu_{ij} \geq 0$ and $\theta_i^{sd}$ are dual variables, $h_i^{sd}(\mathbf{a^{sd}}) = \sum_j a_{ij}^{sd} - \sum_j a_{ji}^{sd}$ and

$$\beta_i^{sd} = \begin{cases} 1, & \text{if } s = i \\ -1, & \text{if } d = i \\ 0, & \text{otherwise} \end{cases} \tag{2.83}$$

First, we minimize over the primary variables.

$$\begin{aligned}
g(\nu, \theta) = \inf_{C_{\max}, a_{ij}^{sd}} \Bigg( C_{\max} &+ \sum_{ij} \nu_{ij} (\sum_{sd} \mu_{sd} a_{ij}^{sd} - C_{\max}) \\
&+ \sum_{s,d,i} \theta_i^{sd} [\beta_i^{sd} - h_i^{sd}(\mathbf{a^{sd}})] \Bigg)
\end{aligned} \tag{2.84}$$

Now, the dual problem can be written as

$$\begin{aligned} d^* = \text{Maximize:} \quad & g(\nu,\theta) \\ \text{Subject To:} \quad & \nu_{ij} \geq 0 \qquad \forall (i,j) \\ & \theta_i^{sd} \quad \text{free} \qquad \forall i,(s,d) \end{aligned} \tag{2.85}$$

By the min-max theorem and by strong duality for linear programs, there is zero duality gap between the dual and primal problems.

$$C^*_{\max}(0) = d^*(0) = \min_{C_{\max},a_{ij}^{sd}} \max_{\nu,\theta} L(\nu,\theta,C_{\max},a) \tag{2.86}$$

$$= \min_{C_{\max},a_{ij}^{sd}} \Big( C_{\max} + \sum_{ij} \nu^*_{ij}(0)(\sum_{sd} \mu_{sd} a_{ij}^{sd} - C_{\max}) + \sum_{s,d,i} \theta_i^{*sd}(0)[\beta_i^{sd} - h_i^{sd}(\mathbf{a^{sd}})] \Big) \tag{2.87}$$

Consider the point $(C^*_{\max}(\sigma), a^*(\sigma))$, which is the optimal routing and max link capacity under traffic with variance $\sigma^2$ over the fixed topology $T$. $(C^*_{\max}(\sigma), a^*(\sigma))$ is obviously feasible and since it is potentially sub-optimal for the deterministic problem, it satisfies

$$\begin{aligned} C^*_{\max}(0) \leq C^*_{\max}(\sigma) &+ \sum_{ij} \nu^*_{ij}(0) \Big( \sum_{sd} \mu_{sd} a_{ij}^{*sd}(\sigma) - C^*_{\max}(\sigma) \Big) \\ &+ \sum_{s,d,i} \theta_i^{*sd}(0)[\beta_i^{sd} - h_i^{sd}(\mathbf{a^{*sd}}(\sigma))] \end{aligned} \tag{2.88}$$

Since the routing above is feasible, $\beta_i^{sd} = h_i^{sd}(\mathbf{a^{*sd}}(\sigma))$ for all $i,(s,d)$. Furthermore, for each link $(i,j)$, the point $(C^*_{\max}(\sigma), a^*(\sigma))$ satisfies

$$C^*_{\max}(\sigma) \geq \sum_{sd} \mu_{sd} a_{ij}^{*sd}(\sigma) + \kappa\sigma \sqrt{\sum_{sd} (a_{ij}^{*sd}(\sigma))^2} \tag{2.89}$$

By combining (2.89) with (2.88), we get the following inequality.

$$C^*_{\max}(0) \leq C^*_{\max}(\sigma) - \kappa\sigma \sum_{ij} \nu^*_{ij}(0) \sqrt{\sum_{sd} (a^{*sd}_{ij}(\sigma))^2} \tag{2.90}$$

$$C^*_{\max}(\sigma) \geq C^*_{\max}(0) + \kappa\sigma \sum_{ij} \nu^*_{ij}(0) \sqrt{\sum_{sd} (a^{*sd}_{ij}(\sigma))^2} \tag{2.91}$$

The above result holds for all topologies, so we consider the specific topology $T_\sigma$. Let $X = \sum_{ij} \nu^*_{ij}(0)\sqrt{\sum_{sd}(a^{*sd}_{ij}(\sigma))^2}$, then equation (2.91) becomes

$$\begin{aligned} C^{T_\sigma,\sigma}_{\max} &\geq C^{T_\sigma,0}_{\max} + \kappa\sigma X && (2.92) \\ &= C^{T_0,0}_{\max} + \Delta + \kappa\sigma X && (2.93) \end{aligned}$$

where equation (2.93) results from the assumption in (2.77). Assume $\Delta + \kappa\sigma X > \kappa\sigma\sqrt{D}$, which is valid if

$$\sigma < \frac{\Delta}{\kappa(\sqrt{D} - X)} = \sigma_0. \tag{2.94}$$

Note that $\sigma_0 \geq 0$ since $0 < X < \sqrt{D}$. Therefore, assuming $\sigma < \sigma_0$ implies

$$\begin{aligned} C^{T_\sigma,\sigma}_{\max} &> C^{T_0,0}_{\max} + \kappa\sigma\sqrt{D} \\ &\geq C^{T_0,\sigma}_{\max} && (2.95) \end{aligned}$$

Equation (2.95) follows from equation (2.80), and is a contradiction, since $T_\sigma$ is optimal for stochastic traffic. Therefore, the assumption made in (2.76) is false and $C^{T_0,0}_{\max} \geq C^{T_\sigma,0}_{\max}$. However, since $T_0$ is optimal for deterministic traffic,

$$C^{T_0,0}_{\max} = C^{T_\sigma,0}_{\max}. \tag{2.96}$$

□

The above theorem can be generalized to the case where the variance on each link is different with a similar "small" $\sigma$ assumption necessary on the vector $\sigma$. While this particular proof doesn't hold for general large values of sigma, the intuition for Conjecture 1 suggests that it is valid.

### 2.5.2 Network Design Implementation

There are techniques for computing the optimal topology for stochastic traffic, provided the network size is small. For large networks, near-optimal formulations and heuristics are required.

#### MILP Formulations

First, we modify the existing linear and integer formulations to include additional variables and constraints for link placement. Let $b_{ij} = 1$ if a link exists between nodes $i$ and $j$, and $b_{ij} = 0$ otherwise. The complete formulation is given in formulation (2.97). Node degree constraints are added, where $\Delta_o$ and $\Delta_i$ are the out-degree and in-degree respectively. These constraints can be replaced with a constraint limiting the maximum number of links, or potentially other design restrictions. This formulation however is intractable for even networks of a moderate size, due to the large number of integer variables and constraints.

To improve this approach, we add the design variables and constraints to the iterative heuristics developed in Section 2.4.4 instead of the MILP. Recall that each iteration of ICA and ACA required solving an LP. By adding the same constraints to those LPs as we added to the formulation in (2.97) for link placement, the iterative schemes can be adapted to solve the link placement problem as well. These iterative algorithms are quicker to execute than the formulation in (2.97), but each iteration now consists of an ILP, and is thus still intractable for large networks.

#### Optimal Techniques for Network Design

Since the formulation in (2.9) calculates the optimal routing and capacity assignment for any given topology, we can exhaustively search through every possible topology

$$
\begin{aligned}
\text{Minimize} \quad & C_{\max} \\
\text{Subject To:} \quad & C_{ij} \geq \sum_{sd \in \mathcal{D}} \mu_{sd} a_{ij}^{sd} + \sqrt{-2 \log \epsilon} \sum_{m=0}^{D} \sqrt{m} y_{ij}^{m} \qquad \forall (i,j) \\
& \sum_{j} a_{ij}^{sd} - \sum_{j} a_{ji}^{sd} = \begin{cases} 1, & \text{if } s = i \\ -1, & \text{if } d = i \\ 0, & \text{otherwise} \end{cases} \qquad \forall (s,d), i \\
& C_{ij} \leq C_{max} \qquad \forall (i,j) \\
& f_{ij}^{sd} \geq a_{ij}^{sd} \qquad \forall (i,j), (s,d) \\
& b_{ij} \geq a_{ij}^{sd} \qquad \forall (i,j), (s,d) \\
& \sum_{sd} f_{ij}^{sd} = \sum_{m=0}^{D} m x_{ij}^{m} \qquad \forall (i,j) \\
& \sum_{m=0}^{D} x_{ij}^{m} = 1 \qquad \forall (i,j) \\
& y_{ij}^{m} \geq \sigma_{sd} a_{ij}^{sd} + M(x_{ij}^{m} - 1) \qquad \forall (s,d), (i,j), m \\
& y_{ij}^{m} \leq M x_{ij}^{m} \qquad \forall (i,j), m \\
& y_{ij}^{m} \geq 0 \qquad \forall (i,j), m \\
& a_{ij}^{sd} \geq 0 \quad \forall i, j, s, d \\
& f_{ij}^{sd}, x_{ij}^{m}, b_{ij} \in \{0, 1\} \qquad \forall (s,d), (i,j), m \\
& \sum_{i} b_{ij} = \Delta_{i} \qquad \forall j \\
& \sum_{j} b_{ij} = \Delta_{o} \qquad \forall i
\end{aligned}
\tag{2.97}
$$

for that with the smallest max link capacity. Every possible permutation of links that meet the design requirements must be enumerated, and the convex routing formulation in (2.9) is applied to each. This is obviously inefficient, but returns the optimal topology.

For example, consider an i.i.d. set of 11 demands over a five node network, where each demand is normally distributed with mean 100 and variance $\sigma^2$. Numbering the nodes from 1 to 5, assume traffic demands exists between the following node pairs: $\mathcal{D} = \{(1,2),(1,3),(2,1),(2,3),(2,4),(2,5),(3,2),(3,4),(3,5),(4,1),(4,5)\}$. The design constraint is that no more than 10 links may be used, with no parallel links. The topologies that are found to be optimal when $\sigma = 0$ are shown in Table 2.1. Then, the same technique is used for the traffic when $\sigma = 10$ and $\sigma = 30$. In both stochastic cases, there is only one optimal topology, namely topology 4 in Table 2.1.

| Topology Number | Links |
|---|---|
| 1 | (1,2),(1,3),(2,1),(2,4),(2,5),(3,2),(3,4),(3,5),(4,1),(4,3) |
| 2 | (1,2),(1,3),(2,1),(2,4),(2,5),(3,2),(3,4),(4,1),(4,3),(4,5) |
| 3 | (1,3),(1,4),(2,1),(2,3),(2,5),(3,2),(3,4),(3,5),(4,1),(4,2) |
| 4 | (1,2),(1,3),(2,1),(2,3),(2,4),(3,2),(3,4),(3,5),(4,1),(4,5) |
| 5 | (1,2),(1,3),(2,1),(2,4),(2,5),(3,1),(3,2),(3,4),(4,3),(4,5) |
| 6 | (1,2),(1,4),(2,1),(2,3),(2,5),(3,2),(3,4),(3,5),(4,1),(4,3) |
| 7 | (1,2),(1,3),(2,3),(2,4),(2,5),(3,1),(3,2),(3,4),(4,1),(4,5) |
| 8 | (1,2),(1,3),(2,1),(2,3),(2,4),(2,5),(3,2),(3,4),(4,1),(4,5) |

Table 2.1: Optimal topologies for deterministic traffic for the demands in $\mathcal{D}$

The optimal stochastic topology is the same for $\sigma = 10$ and $\sigma = 30$ and is a subset of the optimal deterministic topologies, which agrees with Conjecture 1. By assuming that the optimal topologies for stochastic traffic is *always* a subset of the optimal topologies for deterministic traffic, we can propose a new method of finding the optimal stochastic topologies.

Initially, the optimal deterministic topology is found for the set of nodes and demands provided, using existing methods such as the MILP in Section 1.2.1. Once this topology is found, represented by variables $b_{ij}^{(1)}$, the following constraint is added to the deterministic problem.

$$\sum_{\{(i,j)|b_{ij}^{(1)}=0\}} b_{ij} + \sum_{\{(i,j)|b_{ij}^{(1)}=1\}} (1 - b_{ij}) \geq 1 \tag{2.98}$$

This constraint enforces any feasible solution of the current optimization problem ($b_{ij}$) to differ from the solution already obtained ($b_{ij}^{(1)}$) by at least one link. If the solution to the modified MILP has the same objective function as the solution to the original problem, then the new topology is also an optimal topology for deterministic traffic. This is repeated by adding an additional constraint with every topology found, until the objective value of the modified program is larger than the optimal solution, implying that all the optimal deterministic topologies have been found. By assumption, we can exhaustively search all the deterministically optimal topologies using the formulation in (2.9) for the optimal stochastic topologies.

This approach is limited by the execution time of the MILP, which can be high for large networks. Additionally, networks can have thousands of optimal topologies for deterministic traffic.

### Simulated Annealing

Simulated annealing (SA) is a random search heuristic which can be used to find near optimal solutions to optimization problems. The algorithm begins with an arbitrary feasible solution, and a cost computed with respect to an objective function. Then, a random perturbation is applied to the solution, and the cost is reevaluated. The new solution is probabilistically accepted based on the relationship between the two costs. A positive probability of moving to a worse solution avoids the problem of being trapped in a local minima.

Simulated annealing (SA) can be used in this context to compute the link placements. We assume there is a constraint on the maximum number of links. From any feasible network topology, the optimization problem in (2.9) is solved to find $C_{\max}$. Then, a random perturbation is applied to the topology. Specifically, a link $(i, j)$ in the topology and a link $(k, l)$ not in the topology are randomly chosen uniformly from all the links in their respective sets, and link $(k, l)$ is added to the topology while link

$(i, j)$ is removed.

For the new topology, the problem in (2.9) is solved, where the value of the objective function at the solution is $C'_{\max}$. If $C'_{\max} < C_{\max}$, the new topology allows for a smaller maximum link capacity and is accepted unconditionally. If the new topology is worse, it is accepted with probability $q$, where

$$q = \exp\left(\frac{C_{\max} - C'_{\max}}{T}\right) \tag{2.99}$$

The probability of acceptance is a function of the difference between solutions, so that topologies that are much worse are less likely to be accepted than topologies which are only slightly worse. The parameter $T$ is referred to as the *temperature*, borrowing terminology from physics literature. Initially, the temperature of the system is large, such that worse topologies are still likely to be accepted. As the algorithm progresses, temperature is lowered slowly, so less "bad" topologies will be accepted. Specifically, SA iterates for a fixed number of perturbations to simulate reaching a steady state, and then temperature is modified according to $T' = \rho T$ for some $0 \leq \rho \leq 1$. The algorithm is stopped when the probability of escape from some topology is small enough that new topologies are no longer considered.

The SA algorithm should not depend on the initial topology used to begin the search, since SA has measures to protect itself from local minima. Therefore, $M$ links are randomly chosen initially, where $M$ corresponds to the design requirement on the maximum number of links. If the initial links can not support the demands, the topology is labeled as infeasible. As long as the topology is infeasible, a new topology is accepted with probability 1.

### 2.5.3 Network Design Simulation Results

In the previous sections, we have shown that there exist optimal topologies for stochastic traffic that are potentially different from optimal topologies for deterministic traffic. We would like to quantize this difference.

As an example, consider a six node network with the i.i.d demands between the

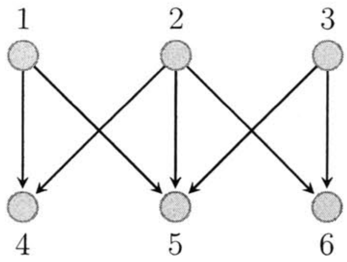
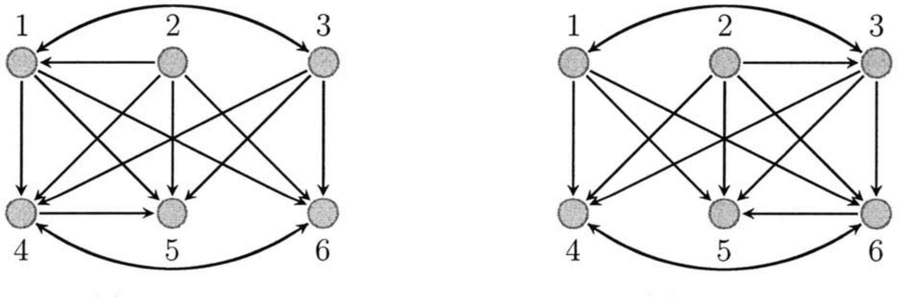


Figure 2-13: Illustration of demands throughout a six-node cluster. A directed edge represents a demand in that direction. All demands are i.i.d.

set of node pairs shown in Figure 2-13. Our goal will be to place up to 15 links between these six nodes, such that the traffic can be optimally routed. Traffic demands are normally distributed with mean 100 and variance $\sigma^2$. The optimal topology is found by first locating the set of optimal topologies for deterministic traffic, and then pruning that set for the optimal stochastic topology for $\sigma = 35$. The two optimal stochastic topologies result in $C_{\max} = 123.89$. These two topologies were chosen from a set of 264 deterministically optimal topologies. Note that each ILP for deterministic design takes between 2 and 3 seconds to solve, whereas each convex optimization takes between 1.5 and 2 seconds. The two optimal topologies are shown in Figure 2-14.



(a) Topology 1

(b) Topology 2

Figure 2-14: The two optimal topologies with 15 links for the demand pattern in 2-13.

By finding the optimal topologies rather than an arbitrary deterministically optimal topology, we save on max link capacity. The stochastically optimal topologies save 8% of the max link capacity that would be necessary for the worst case choice of deterministic topology. On average, the stochastic topologies save 5% of the max link capacity. We can expect to see larger numbers for bigger networks although this

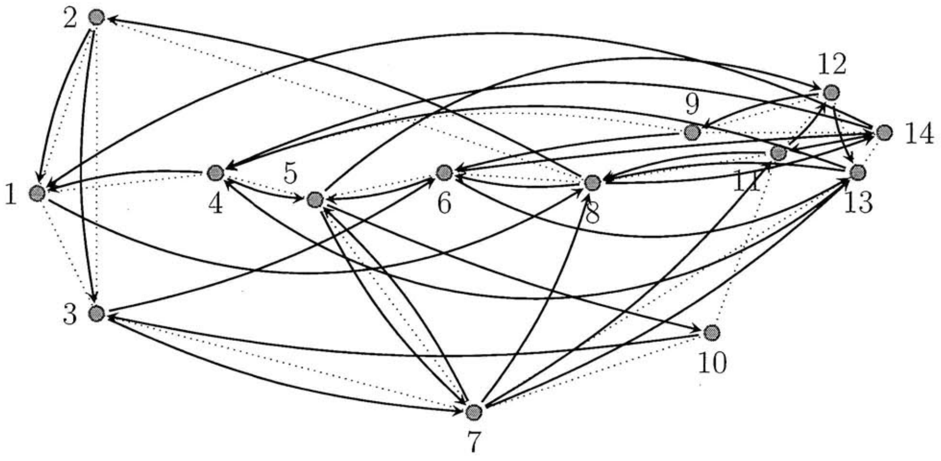


Figure 2-15: Resulting 32-link network (solid) for i.i.d $\mathcal{N}(100, 35^2)$ demands following NSFNET (dotted). The network is designed for $\epsilon = 0.01$, using Simulated Annealing for the link placement, and ICA for the capacity allocation and routing.

is difficult to verify due to the difficulty of solving large ILP's. Lastly, it is worthwhile noting how the ICA with the integer modification performs for this problem. The resulting topology has $C_{\max} = 142.74$. This is substantially worse than optimal, including any deterministically optimal network.

For larger networks, the approach of finding all optimal topologies for deterministic traffic, then searching through those topologies for that which is optimal for stochastic traffic is computationally intensive. In order to design networks for these instances, the simulated annealing approach is used to calculate the link placements, and the routing and capacity allocation problems can be solved using the ICA heuristic developed in Section 3.3.3.

Consider the NSFNET in Figure 2-4. Assume each bidirectional link corresponds to a demand normally distributed with mean 100 and standard deviation 35. The goal of this simulation is to place 32 links on a 14 node network to best route those demands. This can be thought of as redesigning the NSFNET backbone with fewer directed links. The resulting network is shown in Figure 2-15, and requires a maximum link capacity of 456.162. Note that this network may not be optimal, due to the suboptimalities of the simulated annealing approach and the ICA heuristic.

## 2.6 Conclusions

The logical topology design problem for stochastic traffic is fundamentally different than the traditional problem for deterministic traffic. We have presented several formulations for the routing of stochastic traffic and provisioning of capacity in optical networks. The optimal formulation is non-linear, preventing it from being used in discrete optimization problems. Therefore, we extended the results of [4] to unbounded, Gaussian random variables to obtain an ILP for the complete design problem.

The ILPs presented in this work are intractable for networks of moderate size, and so the accuracy of the results for large networks depends on the quality of heuristics. We presented routing heuristics that perform well for certain classes of networks (with high degrees of resource sharing available), but not others. However, the heuristics performed poorly when extended to the link placement sub-problem. Therefore, it would be interesting to develop more accurate heuristics for the design problem. This would allow the difference between stochastic and deterministic designs to be accurately measured, and assess the need for separate network designs considering stochastic traffic.

Chapter 3

# Backup Network Design for Survivability Against Random Network Failures

This chapter presents a framework for providing protection from random link failures as explained in 1.3. We consider the problem of designing a dedicated backup network to provide protection for a primary network subject to independent failures. We present an ILP for the design of such networks using tools from robust optimization, and study their dependence on the primary network reliability.

## 3.1 Failure Model and Problem Statement

Consider a primary network made up of a set of nodes $\mathcal{V}$ and a set of directed links $\mathcal{L}$ connecting these nodes. We assume throughout that the links are directed, as the undirected case is a specific instance of the directed link case.

Each link $(s, d) \in \mathcal{L}$ has a given primary link capacity $C_{sd}^P$, and a positive probability of failure $p$, independent of all other links. Let the random variables $X_{sd}$ equal 1 if link $(s, d)$ fails and 0 otherwise. 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. This model has been adopted by several previous works [30, 33, 37, 38].

A backup network is constructed over the same set of nodes $\mathcal{V}$ and a new set of links $\mathcal{L}_B$, by routing a backup path for each primary link over the backup network and allocating capacity to every backup link. We assume that $\mathcal{L}_B$ can consist only of links $(i, j)$ if there is a primary link connecting nodes $i$ and $j$. An example backup network is shown in Figure 3-1. Furthermore, the backup links are designed such that failures can only occur in the primary network. For each primary link $(s, d) \in \mathcal{L}$, a path on the backup network is chosen such that in the event that $(s, d)$ fails, the traffic load on $(s, d)$ is rerouted over the backup path. Let $b_{ij}^{sd} = 1$ if link $(s, d) \in \mathcal{L}$ uses backup link $(i, j) \in \mathcal{L}_B$ in its backup path. Hence, $b^{sd} = \{b_{ij}^{sd} | \forall(i, j) \in \mathcal{L}_B\}$ represents the backup path for the primary link $(s, d) \in \mathcal{L}$.

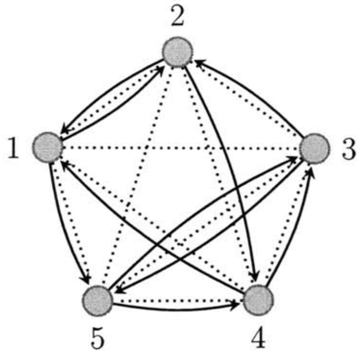


Figure 3-1: Example backup network shown as solid directed links over dotted bidirectional primary network.

A capacity $C_{ij}^B$ is allocated to each backup link $(i, j) \in \mathcal{L}_B$ such that $(i, j)$ can support the increased load due to a random failure scenario with probability $1 - \epsilon$, where $\epsilon > 0$ is a design parameter. Naturally, as $\epsilon$ becomes smaller, more capacity is required on the backup network. Throughout this work we only consider the case where $p \geq \epsilon$, since no backup capacity is required for $p < \epsilon$.

Each primary link has exactly one path in the backup network for protection, and the links in this path can be shared amongst backup paths for multiple primary links. The goal is to construct a minimal cost dedicated backup network as follows:

Minimize:

$$\sum_{(i,j) \in \mathcal{L}_B} C_{ij}^B \tag{3.1}$$

Subject To:

$$\mathbf{P}\Big(\sum_{(s,d)\in\mathcal{L}} X_{sd} b_{ij}^{sd} C_{sd}^{P} > C_{ij}^{B}\Big) \leq \epsilon \qquad \forall (i,j) \in \mathcal{L}_B \tag{3.2}$$

$$\sum_{j} b_{ij}^{sd} - \sum_{j} b_{ji}^{sd} = \begin{cases} 1, & \text{if } s = i \\ -1, & \text{if } d = i \\ 0, & \text{o.w.} \end{cases} \qquad \forall (s,d) \in \mathcal{L}, i \in \mathcal{V} \tag{3.3}$$

$$b_{ij}^{sd}, \in \{0,1\} \qquad \forall (s,d) \in \mathcal{L}, (i,j) \in \mathcal{L}_B \tag{3.4}$$

The constraint in (3.3) is a standard flow conservation constraint for the routing of a single backup path for each primary link. The probabilistic constraint (3.2) is the capacity constraint, restricting the probability that the load on $(i, j)$ due to failures exceeds the backup capacity provisioned on $(i, j)$. This survivability metric, which considers the reliability of each backup link independently, is referred to as the backup-link survivability metric.

### 3.1.1 Probabilistic Survivability Metrics

There are a number of possible survivability metrics that can be considered in this setting; the choice of which will impact the network design. Backup-link survivability considers the reliability of each backup link independently. This metric was introduced in (3.2) and will be the focus of this thesis.

Alternatively, one can consider survivability from a primary link perspective. In particular, a primary link $(s, d)$ is *unprotected* if its capacity cannot be routed from $i$ to $j$ using either the primary or the backup networks due to a failure and insufficient backup capacity. The primary-link reliability constraint is written as

$$\mathbf{P}\big((s,d) \text{ unprotected}\big) = \mathbf{P}\big((s,d) \text{ unprotected} | X_{s,d} = 1\big) \cdot \mathbf{P}\big(X_{sd} = 1\big) < \epsilon \tag{3.5}$$

Assume that if the capacity of backup link $(i,j)$ is insufficient, none of the primary links using that link in their backup paths can be protected.

$$\mathbf{P}\big((s,d)\text{ unprotected}\big) = p{\cdot}\mathbf{P}\Bigg(\bigcup_{(i,j)\in\{(i,j)\in\mathcal{L}_B | b_{ij}^{sd}=1\}} \{(i,j)\text{ has insufficient capacity}\} | X_{sd}=1\Bigg) \tag{3.6}$$

Using the union bound,

$$\begin{aligned} &\leq p\cdot \sum_{(ij)\in\mathcal{L}_B} b_{ij}^{sd}\mathbf{P}\big(\{(i,j)\text{ has insufficient capacty}\}|X_{sd}=1\big) && (3.7)\\ &= p\cdot \sum_{(ij)\in\mathcal{L}_B} b_{ij}^{sd}\mathbf{P}\Bigg(\sum_{(k,l)\in\mathcal{L}\setminus(s,d)} X_{kl}b_{ij}^{kl}C_{kl}^{P} > C_{ij}^{B} - C_{sd}^{P}\Bigg) && (3.8)\end{aligned}$$

Thus, the primary-link survivability constraint can be reduced to the same form as the backup link constraint in (3.2).

One can also consider a survivability constraint on the entire backup network, rather than on each backup link independently.

$$\begin{aligned} \mathbf{P}(\text{backup network fails}) &= \mathbf{P}\Bigg(\bigcup_{(i,j)\in\mathcal{L}_B}\{(i,j)\text{ overflows}\}\Bigg) && (3.9)\\ &\leq \sum_{(i,j)\in\mathcal{L}_B}\mathbf{P}(\{(i,j)\text{ overflows}\}) && (3.10)\end{aligned}$$

Equation (3.10) uses the union bound to express the network-wide survivability constraint in terms of the backup-link survivability. Thus, satisfying the backup link constraint with probability $\frac{\epsilon}{|\mathcal{L}_B|}$ for each backup link is sufficient to satisfy a backup-network constraint with probability $\epsilon$. Consequently, networks designed using the backup network constraint will utilize more resource sharing amongst backup paths, as large backup networks lead to an increased cost.

Lastly, the survivability constraint can be used to ensure the reliability of the entire primary network. It is straightforward to relate primary-network survivability to primary-link survivability.

$$\mathbf{P}(\text{primary network is unprotected}) = \mathbf{P}\Big(\bigcup_{(s,d)\in\mathcal{L}} \{(s,d) \text{ is unprotected}\}\Big) \tag{3.11}$$

$$\leq \sum_{(s,d)\in\mathcal{L}} \mathbf{P}(\{(s,d) \text{ is unprotected}\}) \tag{3.12}$$

The primary network constraint can be satisfied using constraints of the form of (3.5) for each primary link. Further, those primary link constraints can be written in terms of constraints of the form of (3.2).

Each survivability constraint results in a possibly different optimal backup network for a given probability of link failure. The backup-network constraint will lead to backup networks with fewer backup links, where the primary-link approach results in networks with shorter backup paths. However, since each type of constraint can be written in the form of (3.2), an approach for backup network path routing and capacity allocation that satisfies the capacity constraint in (3.2) can be used to satisfy the three other types of reliability constraints. Therefore, the focus in this thesis is restricted to the backup-link constraint in (3.2).

## 3.2 Uniform-Load Primary Networks

Any primary network can be represented by a fully connected graph, with $C_{sd}^P = 0$ for links that are not in the primary network. However, in order to form an intuitive understanding of the general problem, we first explore the backup-network design problem for the special case where each primary link has unit capacity, i.e. $C_{sd}^P = 1 \quad \forall (s,d) \in \mathcal{L}$. The capacity required on each backup link is dictated by the reliability constraint in (3.2). Let $n_{ij}$ be the number of primary links for which backup link $(i,j)$ is part of the backup path. In other words,

$$n_{ij} = \sum_{(s,d)\in\mathcal{L}} b_{ij}^{sd} \tag{3.13}$$

Let $Y_{ij}$ be a random variable representing the number of failed primary links using

$(i, j)$ as part of their backup paths, i.e.,

$$Y_{ij} = \sum_{(s,d)\in\mathcal{L}} b_{ij}^{sd} X_{sd}. \tag{3.14}$$

Since each $X_{sd}$ is an i.i.d bernoulli random variable with parameter $p$, $Y_{ij}$ is a binomial random variable with parameters $n_{ij}$ and $p$. Furthermore, as all the primary links have unit capacity, equation (3.2) can be rewritten as

$$\mathbf{P}\left(\sum_{(s,d)\in\mathcal{L}} X_{sd} b_{ij}^{sd} C_{sd}^{P} > C_{ij}^{B}\right) = \mathbf{P}\left(Y_{ij} > C_{ij}^{B}\right) \tag{3.15}$$

$$= \sum_{y=\lfloor C_{ij}^{B}\rfloor+1}^{n_{ij}} \binom{n_{ij}}{y} p^{y}(1-p)^{n_{ij}-y} \leq \epsilon \qquad \forall (i,j) \in \mathcal{L}_B \tag{3.16}$$

Equation (3.16) uses the cumulative distribution function (CDF) of the binomial distribution. For each link $(i, j)$, let $\mathrm{G}(n_{ij}, p, \epsilon)$ be the minimum value of $C_{ij}^{B}$ satisfying (3.16). Clearly, the capacity required on a backup link increases with the number of primary links it protects, and it decreases as the probability of failure decreases. Additionally, as $\epsilon$ decreases, more capacity is required on each backup link.

The value of $\mathrm{G}(n_{ij}, p, \epsilon)$ can be computed numerically by iterating through the possible values of $C_{ij}^{B}$ until (3.16) is satisfied. However, the iterative scheme may be computationally difficult for large networks, as the sums of combinations can become large. Therefore, we can bound the probability in (3.16) as

$$\mathbf{P}(Y_{ij} \geq \Gamma) \leq \left(\frac{n_{ij}(1-p)}{n_{ij}-\Gamma}\right)^{n_{ij}-\Gamma} \left(\frac{n_{ij}p}{\Gamma}\right)^{\Gamma} \tag{3.17}$$

for $n_{ij}p \leq \Gamma \leq n_{ij}$. This bound follows from direct application of the Chernoff bound. The bound in (3.17) can be compared with the actual distribution function of the binomial random variable to assess its tightness, as shown in Figure 3-2. The bound is slightly loose, but allows an easier computation of $\mathrm{G}(n_{ij}, p, \epsilon)$. This computation must still be numerical, as this bound does not yield an analytical expression for $\Gamma$. Additional bounds exist for binomial tails from which $\mathrm{G}(n_{ij}, p, \epsilon)$ can be expressed analytically, but these are generally too loose for our purposes, especially in the small $p$ regime.

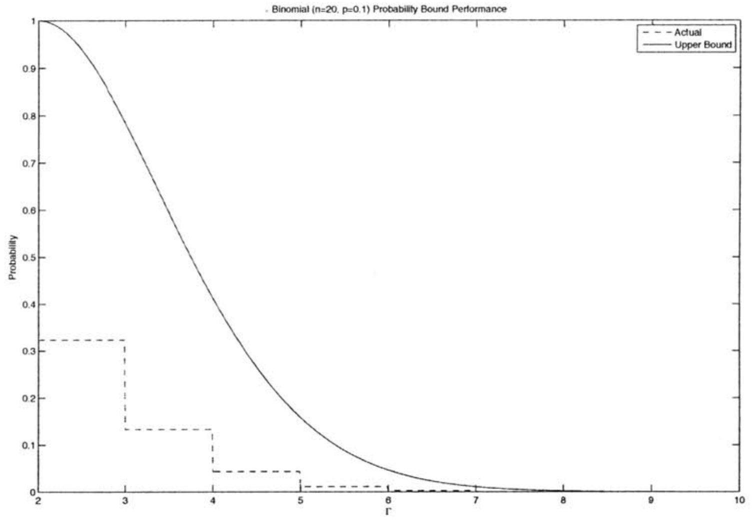
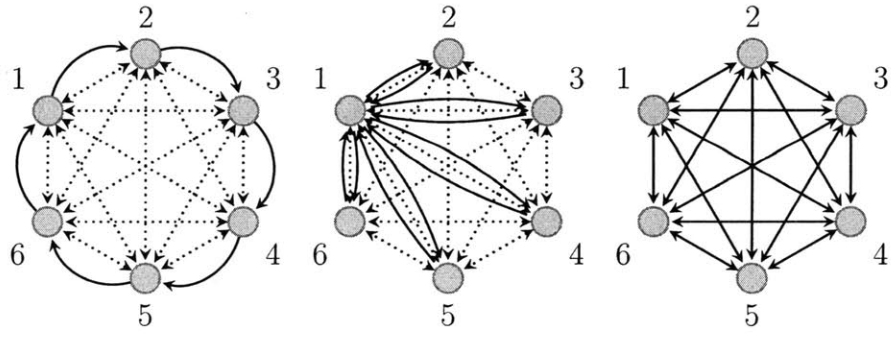


Figure 3-2: Comparing the bound in (3.17) to the actual distribution of the binomial tail. This is for a binomial random variable with parameters $n = 20$ and $p = 0.1$. Note that the bound is only valid for $\Gamma > 2$.

### 3.2.1 Impact of Link Failure Probability



(a) Cycle Protection (b) Two-hop Protection (c) One-hop Protection

Figure 3-3: Sample backup network link placement to protect a 6-node, fully-connected primary network. The dotted lines represent the primary network, and the solid lines represent the backup links.

To gain intuition about the optimal backup network design, we compare three backup routing schemes, shown in Figure 3-3, and show that backup network performance depends on the link failure probability. In the cycle protection scheme of Figure 3-3a, each primary link $(s, d)$ has a backup path lying in a single Hamiltonian cycle through the network. This is the minimum-cost backup network providing protection against a single link failure [1]. Each backup link in this cycle requires unit capacity to protect against a single link failure, resulting in a total cost of $N$ for an $N$-node network. Due to network symmetry, each backup link protects half of the primary links. Therefore, in order to use this scheme to provide protection from a random number of failures with high probability, a total backup capacity of $C^B_{\text{total}} = N \cdot \mathrm{G}(\frac{N(N-1)}{2}, p, \epsilon)$ is required, where $\mathrm{G}(n, p, \epsilon)$ is the smallest value of $C^B_{ij}$ satisfying (3.16).

For large values of $p$, this capacity is $\frac{N^2(N-1)}{2}$, since $\mathrm{G}(n_{ij}, p, \epsilon) = n_{ij}$ for $p$ close to 1. This capacity can be reduced by considering the scheme in Figure 3-3c, where the backup network is a mirror of the primary network, and the backup path for $(s, d)$ is the one-hop path from $s$ to $d$. Since each backup link offers protection to a single primary link, the total capacity required is $C^B_{\text{total}} = N(N-1) \cdot \mathrm{G}(1, p, \epsilon)$. For all values of $p \geq \epsilon$, $C^B_{\text{total}} = N(N-1)$. Thus, the mirror scheme requires a factor of $N$ less capacity than the cycle scheme for primary networks with a high probability of link failure.

While the one-hop protection scheme is optimal in the high-$p$ regime, other schemes are more capacity-efficient for smaller values of $p$. Consider the two-hop scheme in Figure 3-3b, where node 1 serves as a relay node for every backup path. The primary links from node 2 to every other node share the backup link $(2, 1)$, and similarly the primary links from all nodes to node 2 share the backup link $(1, 2)$. Extending this to an $N$-node network, each backup link protects $N - 1$ primaries, and there are $2(N - 1)$ backup links. Thus, $C^B_{\text{total}} = 2(N - 1) \cdot \mathrm{G}(N - 1, p, \epsilon)$.

The three aforementioned routing schemes are compared in Figure 3-4 for a fully connected network with 50 nodes and varying probability of link failure. The cycle-protection scheme, which is optimal in the single failure scenario, requires excessive capacity when considering multiple failures. For values of $p$ close to $\epsilon$, the two-hop routing strategy outperforms the other two strategies. Once $p$ exceeds roughly 0.4, there is no longer a benefit to sharing backup resources, and the one-hop starts to outperform the two-hop schemes. Hence, it is clear that the optimal backup network topology depends on the reliability of the primary network. This is further analyzed in Section 3.3 where the problem is formulated for general primary link capacities.

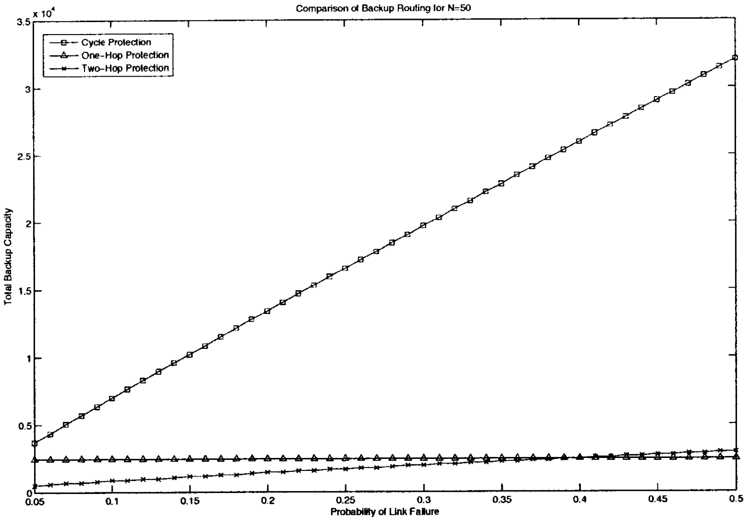


Figure 3-4: Comparison of three protection schemes for an $N - 50$ fully-connected network with unit load.

To illustrate the difference between the backup-link and backup network reliability constraints, Figure 3-5 compares the three routing schemes and the required capacity to satisfy the backup-network constraint of (3.10) with probability $\epsilon = 0.05$. By comparing Figures 3-4 and 3-5, we see that for any given $p$, more capacity is required to meet the network-wide constraint than the link-based constraint. For example, when $p = 0.3$, the two-hop strategy outperforms the one-hop strategy under the backup-link constraint, and vice-versa under the backup-network constraint.

### 3.2.2 Scaling Properties of Backup Network Capacity

Consider the cost of the backup network with respect to that of the primary network. Let $\rho$ be defined as

$$\rho \triangleq \frac{\sum_{(i,j)\in\mathcal{L}_B} C_{ij}^B}{\sum_{(i,j)\in\mathcal{L}} C_{ij}^P}. \tag{3.18}$$

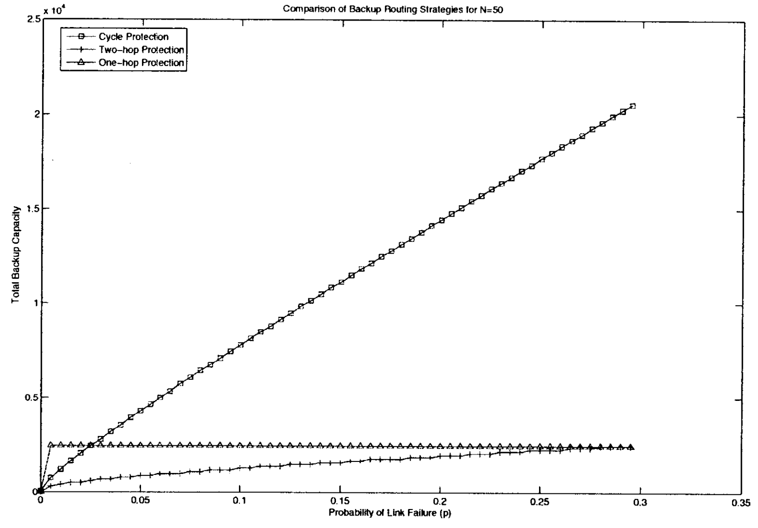


Figure 3-5: Comparison of three protection schemes for an $N-50$ fully-connected network with unit load and backup-network survivability constraint.

I.e., $\rho$ is the ratio of the total capacity of the optimal backup network to that of the primary network. In [1], the authors show that this ratio tends to 0 asymptotically as the network size gets very large for specific networks and single-failure protection. For fully-connected, uniform-load networks, the optimal backup network under single-failure protection is shown in Figure 3-3a, and for this topology

$$\rho = \frac{N}{N(N-1)} = \frac{1}{N-1}. \tag{3.19}$$

Conversely, for protection against random failures, the ratio in (3.18) can be upper bounded using the following proposition.

**Proposition 1.** *Assuming a fully-connected primary network with unit-capacity on each link and probability of link failure p, the ratio between the total capacity of the optimal backup network and that of the primary network can be upper bounded as the primary network size grows large by the following:*

$$\rho \leq 2p \tag{3.20}$$

*Proof.* The optimal total backup capacity is bounded by that of the two-hop scheme considered in Figure 3-3b.

$$\rho = \frac{\sum_{(i,j)\in\mathcal{L}_B} C_{ij}^B}{\sum_{(i,j)\in\mathcal{L}} C_{ij}^P} \leq \frac{2(N-1)\mathrm{G}(N-1,p,\epsilon)}{N(N-1)} \tag{3.21}$$

Consider the behavior of $G(n,p,\epsilon)$ when $n$ is large. Recall that $G(n,p,\epsilon)$ is the required number of primary links out of $n$ that need to be protected to ensure a probability of error of $\epsilon$. Fix a $\delta \geq 0$, and by the weak law of large numbers (WLLN),

$$\lim_{n\to\infty}\left[\mathbf{P}\left(\left|\frac{1}{n}\sum_{m=0}^{n} X_m - p\right| > \delta\right)\right] = 0 \qquad \forall\delta > 0 \tag{3.22}$$

$$\Longrightarrow \lim_{n\to\infty}\left[\mathbf{P}\left(\sum_{m=0}^{n} X_m > n(p+\delta)\right)\right] = 0 \qquad \forall\delta > 0 \tag{3.23}$$

Therefore, as $n$ gets large, $G(n,p,\epsilon) = n(p+\delta)$ is sufficient to meet the probability requirements (for any positive $\epsilon$). In the limit of large n, the inequality in (3.21) reduces to

$$\rho = \frac{\sum_{(i,j)\in\mathcal{L}_B} C_{ij}^B}{\sum_{(i,j)\in\mathcal{L}} C_{ij}^P} \leq \frac{2(N-1)Np}{N(N-1)} = 2p \tag{3.24}$$

□

Therefore, the size of the backup network is a small fraction of the size of the primary network, since $p$ is usually small. Consequently, a backup network designed using the backup-link survivability constraint is a low-cost method of providing protection against random failures in addition to single-link failures. This result is consistent with [1], in that as the primary network size grows large, $p$ approaches zero under the single-failure model.

## 3.3 General-Load Networks

Next, we develop a formulation for general primary link loads. First, we apply the robust optimization results from [4] to formulate a non-linear program for backup capacity provisioning, and develop an equivalent integer linear formulation in terms of new parameters $\Gamma_{ij}$. We show that the choice of these parameters affects the amount of capacity provisioned, and hence the probability of insufficient backup capacity. Then, we add constraints to directly compute these parameters, yielding a solution satisfying the probabilistic constraint in (3.2).

### 3.3.1 Robust Optimization Formulation

In the case of uniform link loads, capacity is allocated to the backup network by computing $\mathrm{G}(n_{ij}, p, \epsilon)$ for each link $(i, j)$. The backup capacity provisioned is the number of primary link failures protected against, as a function of $n_{ij}$, $p$, and $\epsilon$. However, this approach does not apply directly to non-uniform primary link loads, as different links will require different capacities to provide protection. In order to mathematically formulate the problem for general link loads, we will use techniques from the field of robust optimization.

Robust optimization finds a solution to a problem that best fits all possible realizations of the data, when that data is subject to uncertainty. In [4], the authors propose a novel formulation with an adjustable level of conservatism for such problems. Their approach is to introduce an optimization parameter $\Gamma$, and provide sufficient capacity to support all scenarios in which *any* $\Gamma$ of the demands exceed their mean. The solution is guaranteed to be robust for those scenarios, and is shown to be robust for all other scenarios with high probability, determined by $\Gamma$.

A similar approach can be applied to the problem of backup network design for general link loads, where the uncertainty is in the number of primary links that fail. Consider allocating capacity on link $(i, j)$ to protect against any scenario where up to $\Gamma_{ij}$ of the primary links utilizing $(i, j)$ for protection fail. Clearly, for the specific case of uniform loads, the required backup capacity $C_{ij}^B$ is just $\Gamma_{ij}$, and as shown in the previous section, $\Gamma_{ij}$ is given by $\mathrm{G}(n_{ij}, p, \epsilon)$ under the constraint in (3.2). To extend this idea, let $L_{ij}$ be the set of primary links protected by backup link $(i, j)$, i.e. $L_{ij} = \{(s, d) | b_{ij}^{sd} = 1\}$. Let $S_{ij}$ be a set of $\Gamma_{ij}$ primary links in $L_{ij}$ with the largest capacities. Thus, for any $(s, d) \in S_{ij}$, we have

$$C^P_{sd} \geq C^P_{s'd'} \quad \forall (s',d') \in L_{ij} \setminus S_{ij}. \tag{3.25}$$

The backup capacity required to protect against any $\Gamma_{ij}$ primary link failures is given by

$$C^B_{ij} = \sum_{(s,d)\in S_{ij}} C^P_{sd}. \tag{3.26}$$

In a complete form, this constraint can be expressed as

$$C^B_{ij} \geq \max_{S_{ij}|S_{ij}\subseteq\mathcal{L},|S_{ij}|=\Gamma_{ij}} \left\{ \sum_{(s,d)\in S_{ij}} C^P_{sd} b^{sd}_{ij} \right\} \quad \forall (i,j). \tag{3.27}$$

The value of $\Gamma_{ij}$ determines the probability of protection. While $\Gamma_{ij}$ should be chosen such that (3.2) is satisfied, for now we fix the value of $\Gamma_{ij}$ for each link. The capacity constraint in (3.27) replaces the probabilistic constraint in (3.2), leading to the following non-linear optimization problem.

Minimize:

$$\sum_{(i,j)\in\mathcal{L}_B} C^B_{ij}$$

Subject To:

$$\begin{aligned} & C^B_{ij} \geq \max_{S_{ij}|S_{ij}\subseteq\mathcal{L},|S_{ij}|=\Gamma_{ij}} \left\{ \sum_{(s,d)\in S_{ij}} C^P_{sd} b^{sd}_{ij} \right\} \quad \forall (i,j) \\ & \sum_j b^{sd}_{ij} - \sum_j b^{sd}_{ji} = \begin{cases} 1, & \text{if } s = i \\ -1, & \text{if } d = i \\ 0, & \text{o.w.} \end{cases} \quad \forall (s,d) \in \mathcal{L}, i \in \mathcal{V} \\ & b^{sd}_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{L}_B \end{aligned} \tag{3.28}$$

The above is non-linear due to the backup capacity constraint. The problem in (3.28) can be rewritten as a linear program (LP) using duality techniques similar to [4]. For a fixed $b^{sd}_{ij}$ and $\Gamma_{ij}$, the backup capacity of link $(i,j)$ can be written as

$$\beta_{ij}(\mathbf{b_{ij}}, \Gamma_{ij}) = \max_{S_{ij} | S_{ij} \subseteq \mathcal{L}, |S_{ij}| = \Gamma_{ij}} \left\{ \sum_{(s,d) \in S_{ij}} C^P_{sd} b^{sd}_{ij} \right\}, \tag{3.29}$$

which is the solution to the following LP.

$$\begin{aligned} \beta_{ij}(\mathbf{b_{ij}}, \Gamma_{ij}) = \text{maximize} \quad & \sum_{(s,d) \in \mathcal{L}} C^P_{sd} b^{sd}_{ij} z^{sd}_{ij} \\ \text{subject to} \quad & \sum_{(s,d) \in \mathcal{L}} z^{sd}_{ij} \leq \Gamma_{ij} \\ & 0 \leq z^{sd}_{ij} \leq 1 \qquad \forall (s,d) \in \mathcal{L} \end{aligned} \tag{3.30}$$

Assuming the number of primary links $(s,d)$ satisfying $b^{sd}_{ij} = 1$ is larger than or equal to $\Gamma_{ij}$, the LP will choose the $\Gamma_{ij}$ of them with the largest capacities, by setting $z^{sd}_{ij} = 1$ for those links $(s,d)$. This corresponds to choosing the set $S_{ij}$ in (3.26). If there are fewer than $\Gamma_{ij}$ primary links $(s,d)$ satisfying $b^{sd}_{ij} = 1$, then for each of these links $z^{sd}_{ij} = 1$ and the other $(s,d)$ chosen to make $z^{sd}_{ij} = 1$ are irrelevant.

Let $\nu_{ij}$ be the dual variable for the first constraint in (3.30), and let $\theta^{sd}_{ij}$ be the dual variables for the second set of constraints. The dual problem of (3.30) is formulated below.

$$\begin{aligned} \text{minimize} \quad & \nu_{ij}\Gamma_{ij} + \sum_{(s,d) \in \mathcal{L}} \theta^{sd}_{ij} \\ \text{subject to} \quad & \nu_{ij} + \theta^{sd}_{ij} \geq C^P_{sd} b^{sd}_{ij} \quad \forall (s,d) \in \mathcal{L} \\ & \nu_{ij} \geq 0 \\ & \theta^{sd}_{ij} \geq 0 \quad \forall (s,d) \in \mathcal{L} \end{aligned} \tag{3.31}$$

Since there is zero duality gap between problem (3.30) and its dual (3.31), then the optimal value of the objective function in (3.31) is equal to $\beta_{ij}(\mathbf{b_{ij}}, \Gamma_{ij})$. Additionally, since problem (3.28) minimizes $\beta_{ij}(\mathbf{b_{ij}}, \Gamma_{ij})$ for each $(i,j)$, problem (3.31) can be substituted into (3.28) to arrive at the following formulation.

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{(i,j)\in\mathcal{L}_B} C_{ij}^B \\
\text{Subject to} \quad & C_{ij}^B \geq \nu_{ij}\Gamma_{ij} + \sum_{(s,d)\in\mathcal{L}} \theta_{ij}^{sd} \qquad \forall(i,j)\in\mathcal{L}_B \\
& \nu_{ij} + \theta_{ij}^{sd} \geq C_{sd}^P b_{ij}^{sd} \qquad \forall(s,d)\in\mathcal{L},(i,j)\in\mathcal{L}_B \\
& \sum_j b_{ij}^{sd} - \sum_j b_{ji}^{sd} = \begin{cases} 1, & \text{if } s=i \\ -1, & \text{if } d=i \qquad \forall(s,d)\in\mathcal{L}, i\in\mathcal{V} \\ 0, & \text{otherwise} \end{cases} \\
& b_{ij}^{sd}\in\{0,1\} \qquad \forall(i,j)\in\mathcal{L}_B \\
& \nu_{ij},\theta_{ij}^{sd} \geq 0 \qquad \forall(s,d)\in\mathcal{L},(i,j)\in\mathcal{L}_B
\end{aligned} \tag{3.32}
$$

If fewer than $\Gamma_{ij}$ links in $L_{ij}$ fail, the capacity allocated in (3.27) will be sufficient. Therefore, the probability of insufficient backup capacity can be upper bounded using the tail probability of a binomial random variable.

$$
\mathbf{P}\Big(\sum_{(s,d)\in\mathcal{L}} X_{sd} b_{ij}^{sd} C_{sd}^P > C_{ij}^B\Big) \leq \mathbf{P}\Big(Y_{ij} > \Gamma_{ij}\Big) \tag{3.33}
$$

The capacity allocated in (3.26) is sufficient to meet the reliability constraint in (3.33) with probability $\epsilon$ if $\Gamma_{ij} = \mathrm{G}(n_{ij}, p, \epsilon)$. However, $n_{ij}$ is an optimization variable, on which $\Gamma_{ij}$ depends. Thus, the remaining task is to modify (3.32) to directly compute the value of $\Gamma_{ij}$ for each link using an ILP formulation.

### 3.3.2 Complete Formulation

Since $\Gamma_{ij}$ cannot be computed analytically, we create a table a priori in which the $m$th entry $\Gamma(m)$ equals $\mathrm{G}(m, p, \epsilon)$, computed numerically. We develop an ILP that leads to the direct computation of $n_{ij}$ in order to index the table.

To compute $n_{ij}$, let $x_{ij}^m = 1$ if $n_{ij} = m$, and 0 otherwise. The following constraints are introduced.

$$\sum_{m=0}^{N(N-1)} x_{ij}^m = 1 \qquad \forall (i,j) \in \mathcal{L}_B \tag{3.34}$$

Constraint (3.34) enforces $x_{ij}^m$ to be equal to 1 for only one value of $m$ for each backup link.

$$\sum_{(s,d)\in\mathcal{L}} b_{ij}^{sd} = \sum_{m=0}^{N(N-1)} m \cdot x_{ij}^m \qquad \forall (i,j) \in \mathcal{L}_B \tag{3.35}$$

Constraint (3.35) ensures that the number of primary links utilizing a backup link $(i,j)$ is equal to the value of $m$ for which $x_{ij}^m = 1$. Consequently, $\Gamma_{ij}$ can be represented by the following.

$$\Gamma_{ij} = \mathrm{G}(n_{ij}, p, \epsilon) = \sum_{m=0}^{N(N-1)} \Gamma(m) x_{ij}^m \tag{3.36}$$

The capacity constraint of (3.32) is rewritten as

$$C_{ij}^B \geq \sum_{m=0}^{N(N-1)} \Gamma(m) \nu_{ij} x_{ij}^m + \sum_{(s,d)\in\mathcal{L}} \theta_{ij}^{sd} \tag{3.37}$$

Since the product $\nu_{ij} x_{ij}^m$ is non-linear, another set of optimization variables is added to represent this product in linear form. Let $y_{ij}^m$ be a nonnegative variable satisfying the following constraints:

$$y_{ij}^m \geq \nu_{ij} + K(x_{ij}^m - 1) \qquad \forall (i,j), m \tag{3.38}$$

$$y_{ij}^m \leq K x_{ij}^m \qquad \forall (i,j), m \tag{3.39}$$

$$y_{ij}^m \geq 0 \qquad \forall (i,j), m \tag{3.40}$$

In the above equations, $K$ is a large number such that $K > \max_{sd} C_{sd}^P$. When $x_{ij}^m = 0$, then $x_{ij}^m \nu_{ij} = 0$, and constraints (3.39) and (3.40) force $y_{ij}^m$ to 0. On the other hand, if $x_{ij}^m = 1$, constraint (3.38) will force $y_{ij}^m \geq \nu_{ij}$, which at the optimal solution will be satisfied with equality. These constraints lead to an ILP formulation for backup network design, given in (3.41).

The following is an ILP formulation for the design of a dedicated backup network to protect against random failures.

Minimize:

$$\sum_{(i,j)\in\mathcal{L}_B} C_{ij}^B$$

Subject To:

$$\begin{aligned}
&C_{ij}^B \geq \sum_{m=0}^{N(N-1)} y_{ij}^m \Gamma(m) + \sum_{(s,d)\in\mathcal{L}} \theta_{ij}^{sd} \qquad \forall (i,j)\in\mathcal{L}_B \\
&\nu_{ij} + \theta_{ij}^{sd} \geq C_{sd}^P b_{ij}^{sd} \qquad \forall (s,d)\in\mathcal{L}, (i,j)\in\mathcal{L}_B \\
&\sum_{m=0}^{N(N-1)} x_{ij}^m = 1 \qquad \forall (i,j)\in\mathcal{L}_B \\
&\sum_{(s,d)\in\mathcal{L}} b_{ij}^{sd} \leq \sum_{m=0}^{N(N-1)} m\cdot x_{ij}^m \qquad \forall (i,j)\in\mathcal{L}_B \\
&y_{ij} \geq \nu_{ij} + K(x_{ij}^m - 1) \qquad \forall (i,j)\in\mathcal{L}_B, m \\
&y_{ij}^m \leq K x_{ij}^m \qquad \forall (i,j)\in\mathcal{L}_B, m \\
&\sum_j b_{ij}^{sd} - \sum_j b_{ji}^{sd} = \begin{cases} 1, & \text{if } s=i \\ -1, & \text{if } d=i \\ 0, & \text{o.w.} \end{cases} \qquad \forall (s,d)\in\mathcal{L}, i\in\mathcal{V} \\
&b_{ij}^{sd}, x_{ij}^m \in \{0,1\} \qquad \forall (i,j)\in\mathcal{L}_B, m \\
&\theta_{ij}^{sd} \geq 0, \nu_{ij} \geq 0, y_{ij}^m \geq 0 \qquad \forall (s,d)\in\mathcal{L}, (i,j)\in\mathcal{L}_B, m
\end{aligned} \tag{3.41}$$

This formulation calculates the backup paths and capacity allocation for a dedicated backup network satisfying the survivability constraint in (3.2).

### 3.3.3 Simulated Annealing

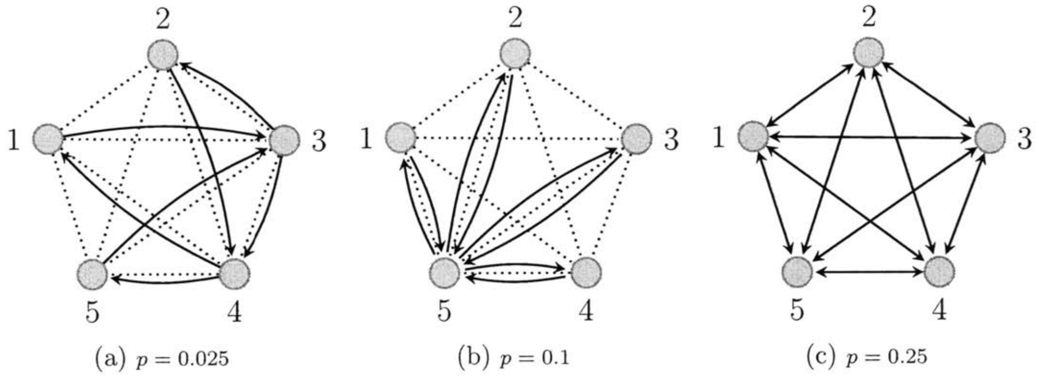
The ILP in (3.41) can be directly solved for small instances, but becomes intractable for large networks. There are a number of heuristic approaches to solving ILPs, such as randomized rounding [27], tabu search [31], and simulated annealing [6]. Here, we employ a simulated annealing approach to estimate the backup path routing in (3.41).

Simulated annealing (SA) is a random search heuristic which can be used to find near optimal solutions to optimization problems. The algorithm begins with an arbitrary feasible solution, and a cost computed with respect to an objective function. Then, a random perturbation is applied to the solution, and the cost is reevaluated. The new solution is probabilistically accepted based on the relationship between the two costs. A positive probability of moving to a worse solution avoids the problem of being trapped in a local minima. SA has been used previously on network survivability problems [34].

For a fixed backup path routing, the computation of the optimal backup capacity $C_{ij}^B$ is straightforward. Therefore, we use simulated annealing to estimate the backup path routing. For the problem in (3.41), the solution at each SA iteration is the backup path for each primary link, and the cost is the total backup capacity, computed using (3.27). Perturbations are applied to this solution by randomly recomputing the backup path for a randomly chosen primary link. The current network with cost $C_{total}^B$ is modified by changing a single backup path, and the network cost $C_{\text{total}}^{B\prime}$ is recomputed. The new backup network is accepted with probability $\max(q, 1)$ where

$$q = \exp\left(\frac{C_{\text{total}}^B - C_{\text{total}}^{B\prime}}{T}\right). \qquad (3.42)$$

Hence, better solutions are unconditionally accepted and worse solutions are accepted with probability $q$. The parameter $T$ in equation (3.42) represents the "temperature" of the system. At high temperatures, there is a high probability of accepting a solution with a larger cost than the current solution. This prevents the algorithm from getting trapped in a local minima. The temperature decreases after a number



(a) $p = 0.025$ (b) $p = 0.1$ (c) $p = 0.25$

Figure 3-6: Optimal backup networks shown as solid links over dotted primary networks for different probabilities of link failure. Designed using $\epsilon = 0.01$.

of iterations depending on the network size by $T' = \rho T$, for $0 < \rho < 1$. SA cannot escape local minima if $\rho$ is too small, but high values of $\rho$ result in long computation times. Eventually, $T$ becomes small enough that the probability of accepting a worse solution approaches zero. At this point, the algorithm terminates and returns the resulting backup network.

The number of total iterations required to converge to an optimal solution, which is a function of the starting value of $T$, the value of $\rho$, and the number of iterations before reducing $T$, depends on the network size. However, this increase in processing time is polynomial, and consequently, solving large networks is tractable using this approach.

## 3.4 Simulation Results

Consider a five-node, fully-connected topology where each primary link has unit-capacity. Due to the small size of this network, the ILP in (3.41) can be solved to compute the optimal backup topologies for different values of $p$. These backup networks are shown in Figure 3-6. For small values of $p$, the backup topology consists of few links, whereas for large values of $p$, the backup network resembles the primary network. Table 3.1 summarizes the results of the backup networks for different values of $p$, using all of the design heuristics discussed. Cycle protection, two-hop protection, and one-hop protection refer to the strategies analyzed in Section 3.2. The optimal column refers to the solution returned by solving the ILP in (3.41) using CPLEX, and the SA column refers to an approach where simulated annealing is used to solve the ILP.

| Strategy | $p = 0.025$ | $p = 0.05$ | $p = 0.075$ | $p = 0.1$ | $p = 0.25$ |
|---|---|---|---|---|---|
| optimal | 7 | 10 | 13 | 16 | 20 |
| cycle | 10 | 15 | 15 | 20 | 30 |
| two-hop | 8 | 16 | 16 | 16 | 24 |
| one-hop | 20 | 20 | 20 | 20 | 20 |
| SA | 7 | 11 | 13 | 16 | 20 |

Table 3.1: Backup network capacity required for topologies designed using different strategies. $\epsilon = 0.01$ in each design.

The table shows that for $p = 0.1$, the two-hop protection scheme is optimal, and for $p = 0.25$, the one-hop protection scheme is optimal. Furthermore, the simulated annealing heuristic performs very close to optimal for different values of $p$. Since the optimal topology depends on the probability of link failure, it is therefore necessary to use a different backup routing scheme depending on the link failure probabilities.

The heuristics can be extended to larger networks, but the ILP in (3.41) cannot be solved directly for large networks. Thus, we use the SA approach to solve the ILP for backup network design for large primary networks.

Consider the NSFNET primary network shown in Figure 2-4. Each link is bidirectional, with unit capacity in each direction. Our goal is to construct a backup network consisting of links $(i, j) \in \mathcal{L}_B$, where $i$ and $j$ are connected by a link in the NSFNET. The survivability constraint in (3.2) must be satisfied with probability $\epsilon = 0.05$. The SA algorithm, shown to be near-optimal for smaller networks, is used to compute the backup network for this larger example. The resulting backup networks for probability of link failure $p = 0.075$ and $p = 0.10$ are shown in Figures 3-7 and 3-8 respectively.

In the backup network of Figure 3-7, a total capacity of 24 is required. Most backup links protect up to 5 primary links. In the case of Figure 3-8, where the probability of link failure is higher, a total capacity of 28 is needed. The backup

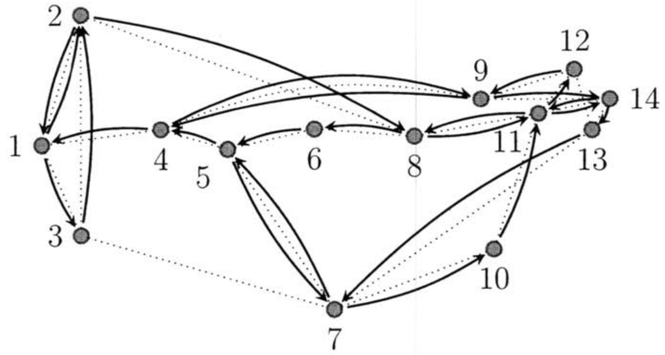


Figure 3-7: Backup network (solid) shown for the NSFNET (dotted) with the restriction that the backup network must be a subset of the primary network. The primary network here assumes a probability of link failure of 0.075, and the backup network is designed for $\epsilon = 0.05$.

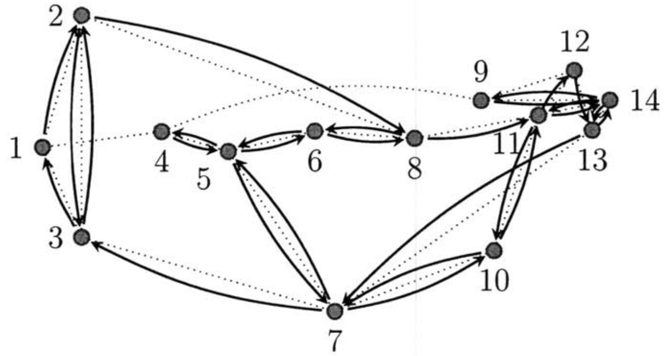


Figure 3-8: Backup network (solid) shown for the NSFNET (dotted) with the restriction that the backup network must be a subset of the primary network. The primary network here assumes a probability of link failure of 0.1, and the backup network is designed for $\epsilon = 0.05$.

links in this example protect an average of 3 primary links. If the probability of link failure increases to $p = 0.25$, the resulting backup topology is a mirror of the primary topology, requiring a capacity of 42. As $p$ increases, the number of backup links rises, and similarly the number of primary links being protected by each backup link falls, until the network follows the one-hop protection scheme. These results are summarized in Table 3.2.

| Link Failure Probability | $\sum_{(i,j)\in\mathcal{L}_B} C_{ij}^B$ | Average $n_{ij}$ |
|---|---|---|
| $p = 0.06$ | 22 | 4.87 |
| $p = 0.075$ | 24 | 4.42 |
| $p = 0.085$ | 27 | 3.59 |
| $p = 0.10$ | 28 | 3 |
| $p = 0.175$ | 34 | 1.88 |
| $p = 0.25$ | 42 | 1 |

Table 3.2: Comparison of backup networks for NSFNET with different probabilities of primary link failure. Networks were designed using $\epsilon = 0.05$. Average $n_{ij}$ refers to the average number of primary links being protected by a backup link.

## 3.5 Conclusions

Dedicated backup networks are a low-cost and efficient method for providing protection against multiple (random) failures. In the event of a failure, the load on the failed link can be automatically rerouted over a predetermined path in the backup network, providing fast recovery from network failures. We formulated the backup network design problem as an ILP for primary networks with general link capacities and independent, identically distributed probabilities of link failure. For primary networks with rare failures, backup networks are shown to use fewer links, with more resource sharing among backup paths. Conversely, when the primary network has a high probability of link failure, the backup network consists of shorter backup paths. For larger primary networks, a simulated annealing approach was presented to solve the backup network design ILP. This approach has been shown to perform near optimally in designing dedicated backup networks. The SA algorithm can be adjusted to trade-off between computation time and accuracy, with computation time increasing polynomially with the network size.

Throughout this work, it has been assumed that the backup network is free from failures. This assumption holds if the backup links are physically designed such that they are more robust to failures. It would be interesting to extend the approach presented in this paper to a failure model in which the backup links are also susceptible to failure.

# Bibliography

[1] M. Banner and A. Orda. Designing low-capacity backup networks for fast restoration. In *Proc. of INFOCOM*. IEEE, 2010.

[2] W. Ben-Ameur and H. Kerivin. Routing of uncertain traffic demands. *Optimization and Engineering*, 6(3):283–313, 2005.

[3] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–14, 1999.

[4] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, pages 35–53, 2004.

[5] R. Bhandari. *Survivable Networks: Algorithms for Diverse Routing*. Kluwer Academic, 1999.

[6] SP Brooks and BJT Morgan. Optimization using simulated annealing. *The Statistician*, 44(2):241–257, 1995.

[7] J. Cao, D. Davis, S.V. Wiel, and B. Yu. Time-varying network tomography: Router link data. *Journal of the American Statistical Association*, 95(452), 2000.

[8] H. Choi, S. Subramaniam, and H.A. Choi. On double-link failure recovery in wdm optical networks. In *IEEE INFOCOM*, volume 2, pages 808–816. Citeseer, 2002.

[9] S. Gast and A. Keimer. Robust optimization in network flows. 2009.

[10] A. Grosso, E. Leonardi, M. Mellia, and A. Nucci. Logical topologies design over wdm wavelength routed networks robust to traffic uncertainties. *IEEE Communications Letters*, 5(4):172–174, 2001.

[11] W.D. Grover. *Mesh-based survivable networks: options and strategies for optical, MPLS, SONET, and ATM Networking*. prentice hall, 2004.

[12] L. Guo, H. Yu, and L. Li. Segment shared protection for survivable meshed wdm optical networks. *Optics communications*, 251(4-6):328–338, 2005.

[13] M. Herzberg, S.J. Bye, and A. Utano. The hop-limit approach for spare-capacity assignment in survivable networks. *IEEE/ACM Transactions on Networking (TON)*, 3(6):784, 1995.

[14] B.G. Jozsa, D. Orincsay, and A. Kern. Surviving multiple network failures using shared backup path protection. In *Proc. ISCC*, volume 1, pages 1333–1340, 2003.

[15] O.E. Karasan, M. Pinar, and H. Yaman. Robust dwdm routing and provisioning under polyhedral demand uncertainty. In *Proc. of INOC*, pages 242–249, 2005.

[16] M. Kodialam, TV Lakshman, and S. Sengupta. Efficient and robust routing of highly variable traffic. In *Third Workshop on Hot Topics in Networks (HotNets-III)*. Citeseer, 2004.

[17] R.M. Krishnaswamy and K.N. Sivarajan. Design of logical topologies: a linear formulation for wavelength-routed optical networks with no wavelength changers. *IEEE/ACM Transactions on Networking (TON)*, 9(2):186–198, 2001.

[18] E. Leonardi, M. Mellia, and M.A. Marsan. Algorithms for the logical topology design in wdm all-optical networks. *Optical Networks Magazine*, 1(1):35–46, 2000.

[19] Y. Liu, D. Tipper, and P. Siripongwutikorn. Approximating optimal spare capacity allocation by successive survivable routing. *IEEE/ACM Trans. Netw.*, 13(1):198–211, 2005.

[20] S.S. Lumetta et al. Capacity versus robustness: a tradeoff for link restoration in mesh networks. *Journal of Lightwave Technology*, 18(12):1765, 2000.

[21] K. Meesublak. Network design under demand uncertainty. In *Proc. of the Asia-Pacific Advanced Network Meeting*, 2008.

[22] D. Mitra and Q. Wang. Stochastic traffic engineering for demand uncertainty and risk-aware network revenue management. *IEEE/ACM Transactions on Networking (TON)*, 13(2):233, 2005.

[23] E. 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):800–809, May 2002.

[24] A.M. Odlyzko. Internet traffic growth: Sources and implications. In *Proc. SPIE*, volume 5247, pages 1–15. Citeseer, 2003.

[25] 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, Sep. 2002.

[26] D. Papadimitriou et al. Inference of shared risk link groups. *Internet Draft.*

[27] P. Raghavan and C.D. Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

[28] S. Ramamurthy, Laxman Sahasrabuddhe, and Biswanath Mukherjee. Survivable wdm mesh networks. volume 21, pages 870–883, 2003.

[29] R. Ramaswami and K.N. Sivarajan. Design of logical topologies for wavelength-routed all-optical networks. In *IEEE INFOCOM*, pages 1316–1316. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 1995.

[30] J. Segovia, E. Calle, P. Vila, J. Marzo, and J. Tapolcai. Topology-focused availability analysis of basic protection schemes in optical transport networks. *Journal of Optical Networking*, 7:351–+, 2008.

[31] C.C. Shyur, T.C. Lu, and U.P. Wen. Applying tabu search to spare capacity planning for network restoration. *Computers & Operations Research*, 26(12):1175–1194, 1999.

[32] AL Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations research*, 21(5):1154–1157, 1973.

[33] M. Tornatore, G. Maier, and A. Pattavina. Availability design of optical transport networks. *IEEE Journal on Selected Areas in Communications*, 23(8):1520–1532, Aug. 2005.

[34] B. Van Caenegem, W. Van Parys, F. De Turck, and PM Demeester. Dimensioning of survivable wdm networks. *IEEE Journal on Selected Areas in Communications*, 16(7):1146–1157, 1998.

[35] I. Widjaja, I. Saniee, A. Elwalid, and D. Mitra. Online traffic engineering with design-based routing. In *ITC Specialist Workshop*.

[36] D. Xu, Y. Xiong, C. Qiao, and G. Li. Trap avoidance and protection schemes in networks with shared risk link groups. *IEEE/OSA Journal of Lightwave Technology*, 21(11):2683–2693, Nov. 2003.

[37] J. Zhang and B. Mukherjee. A review of fault management in wdm mesh networks: Basic concepts and research challenges. *IEEE Network*, pages 41–48, Mar. 2004.

[38] J. Zhang, K. Zhu, H. Zang, and B. Mukherjee. Service provisioning to provide per-connection-based availability guarantee in wdm mesh networks. In *OFC*, Atlanta, GA, Mar. 2003.

[39] R. Zhang-Shen and N. McKeown. Designing a predictable internet backbone network. In *HotNets III*, 2004.

[40] D. Zhou and S. Subramaniam. Survivability in optical networks. *IEEE network*, 14(6):16–23, 2000.