IMPROVING NETWORK RELIABILITY:
ANALYSIS, METHODOLOGY, AND ALGORITHMS

A Dissertation

by

GRAHAM BROADT BOOKER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2010

Major Subject: Computer Engineering

IMPROVING NETWORK RELIABILITY:

ANALYSIS, METHODOLOGY, AND ALGORITHMS

A Dissertation

by

GRAHAM BROADT BOOKER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Alexander Sprintson
Committee Members,	Riccardo Bettati
	Costas Georghiades
	Srinivas Shakkottai
Head of Department,	Costas Georghiades

May 2010

Major Subject: Computer Engineering

ABSTRACT

Improving Network Reliability:

Analysis, Methodology, and Algorithms. (May 2010)

Graham Broadt Booker, B.S., Texas A&M University

Chair of Advisory Committee: Dr. Alexander Sprintson

The reliability of networking and communication systems is vital for the nation's economy and security. Optical and cellular networks have become a critical infrastructure and are indispensable in emergency situations. This dissertation outlines methods for analyzing such infrastructures in the presence of catastrophic failures, such as a hurricane, as well as accidental failures of one or more components. Additionally, it presents a method for protecting against the loss of a single link in a multicast network along with a technique that enables wireless clients to efficiently recover lost data sent by their source through collaborative information exchange.

Analysis of a network's reliability during a natural disaster can be assessed by simulating the conditions in which it is expected to perform. This dissertation conducts the analysis of a cellular infrastructure in the aftermath of a hurricane through Monte-Carlo sampling and presents alternative topologies which reduce resulting loss of calls. While previous research on restoration mechanisms for large-scale networks has mostly focused on handling the failures of single network elements, this dissertation examines the sampling methods used for simulating multiple failures. We present a quick method of finding a lower bound on a network's data loss through enumeration of possible cuts as well as an efficient method of finding a tighter lower bound through genetic algorithms leveraging the *niching* technique.

Mitigation of data losses in a multicast network can be achieved by adding redundancy and employing advanced coding techniques. By using Maximum Rank

Distance (MRD) codes at the source, a provider can create a parity packet which is effectively linearly independent from the source packets such that all packets may be transmitted through the network using the *network coding* technique. This allows all sinks to recover all of the original data even with the failure of an edge within the network. Furthermore, this dissertation presents a method that allows a group of wireless clients to cooperatively recover from erasures (e.g., due to failures) by using the *index coding* techniques.

To my family, friends, and God, whose support helped me through all these years.

ACKNOWLEDGMENTS

I would like to thank my committee chair, Dr. Sprintson, and my committee members, Dr. Shakkottai, Dr. Georghiades, and Dr. Bettati for their guidance and support through my years pursuing a Ph.D. I would also like to thank my family and friends who stood by me through the entire process making my time at Texas A&M more enjoyable.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

Communication networks have become critical to modern life. They are used in day to day activities as well as in emergencies. These networks come in many forms, be it cellular networks allowing mobile voice and data communication as well as backbone networks allowing fast data exchange between stationary nodes. Due to similarities in architecture, these networks face similar challenges such as connectivity, reliability, and latency while serving a user base who simply desires fast, cheap, and reliable communications.

One scenario where users present such demands on a network is the expectation of a functional cellular network in the presence of a disaster. In addition to users' demands, emergency services personnel also need cellular networks to serve as a backup to their own communications equipment as was the case after Hurricane Katrina [1]. These networks are also vulnerable to damage from the same source, causing failures in wireless and wired networks [2]. Therefore, evaluation of the reliability of such a network in the presence of a disaster and methods for mitigating the effects of the damage are important research topics. Such networks can also suffer damage without the presence of a disaster due to random failures.

Users also desire a functioning network in every day usage. Long haul computer networks suffer damage from construction equipment, storms, ships anchors, and other causes. The FCC estimates that a long haul fiber cable suffers from 3 cuts for every 1000 miles every year with the repair averaging 14 hours [3]. Such networks are often over-provisioned or use 1+1 protection to prevent losses from such events [4].

The journal model is *IEEE Transactions on Automatic Control.*

While such protections do reduce the impact of disconnects, numerous links may be cut at the same time leading to degraded performance, as demonstrated in Egypt and India in early 2008 [5]. It is important to be able to analyze such a network in order to determine its weak points and the likelihood of data loss. With such a tool, one can study potential improvements in light of the network's overall reliability. Providers can also employ some techniques to reduce the impact of small network failures.

In some uses of a broadcast network, providers would like to have the ability for a node to recover data immediately without the need for the source to retransmit any data. For example, many users may be collectively viewing a live broadcast where late data is effectively missing data. This can be accomplished through network coding to provide path diversity in a broadcast channel [6, 7]. With this technique, a provider can broadcast data to multiple clients with instantaneous recovery from a single edge failure anywhere within the network without the provider taking any action. Additionally, clients can help each other to recover missing data.

If a set of clients are able to communicate with each other, they can employ cooperative data exchange algorithms to fill in each other's gaps in the data [8]. In several situations, clients may have a cheaper or faster communication with each other than with the source for the data. An example is cellular phones fetching the same data through a cellular tower, but the phones can also communicate with each other through a WiFi connection, which is faster than the cellular link. In this setup, the users of each phone would wish for their phones to make the minimal number of additional transmissions necessary to accomplish the task, so finding the optimal number of broadcast packets is an important problem.

This dissertation is organized as follows. Chapter II studies the cellular network in a virtual city and its performance in the presence of a hurricane. Chapter III presents techniques for analyzing the reliability of long-haul networks with rare but catastrophic failure events. Chapter IV provides a technique to broadcast multiple packets through a network and withstand a single edge failure. Chapter V gives a method where nodes interested in common data may cooperate with each other to fill in gaps in reception without intervention from the data's source. Finally, Chapter VI summarizes these approaches and their significance.

CHAPTER II

ESTIMATING CELLULAR NETWORK PERFORMANCE DURING HURRICANES

A. Introduction

Cellular networks serve a critical role during and immediately after hurricanes. They allow citizens to communicate with emergency services if land-line communication systems have been lost, and they allow families to communicate and organize response plans. Cellular networks have also been used as a backup communication system for emergency personnel when their primary systems fail. For example, during Hurricane Katrina police radio bands failed due to power outages, combined with a lack of fuel for backup generators among other reasons [1]. However, cellular communication systems are not highly reliable during hurricanes. During recent hurricanes, cellular towers have been damaged and backup generators have failed or run out of fuel, rendering the attached communication equipment inoperable [1, 2]. In addition, microwave and optical fiber-optic links between cellular towers and backbone systems have also failed, rendering that portion of the system equally inoperable [2]. Estimating the reliability of cellular networks during hurricanes is an important problem, and methods are needed that incorporate the effects of failures in multiple nodes and links together with the effects of changes in network traffic during and immediately after a hurricane. At the same time, these methods must utilize appropriate models of the reliability of cellular towers, antennae, and support equipment such as backup generators during hurricanes. The building blocks of an approach for estimating cellular network reliability are available; yet developing an accurate and efficient estimation model remains a challenge. The first building block includes structural reliability analysis methods for modeling the performance of cellular communication towers under hurricane wind loadings, fault trees, event trees, and other reliability analysis methods for modeling the performance of individual cellular nodes (e.g. a tower, antennae, an external power source, and a backup power source). The second building block includes traffic models for estimating the subscriber traffic and call arrival rate at each cellular tower. The last building block includes simulation techniques for estimating the performance of the cellular system over wide geographical areas. These techniques should accurately simulate the routing and restoration protocols that determine the network availability for given physical states of the system. In this chapter we show how this interdisciplinary set of approaches can be combined to develop an integrated approach for assessing cellular network reliability during hurricanes. We demonstrate this approach using a synthetic cellular network. This chapter is organized as follows. Section B provides an overview of relevant past work. Section C presents our integrated modeling approach together with the case study system that we use to demonstrate the approach. Section D discusses the results for this case study area. Section E summarizes our approach and its performance.

B. Background

1. Cellular Network Performance and Modeling

A typical cellular network is comprised of a set of cellular towers, each of which covers a specific area [9]. A tower's coverage is roughly hexagonal in shape with a radius of about 6.44 km (4 mi). Towers also communicate with each other via two possible methods. The first is the use of buried fiber-optic links between the tower and the local switching station. This method is the most reliable and provides the ability to carry the most calls, but it is also the most expensive. A less expensive alternative is the use of microwave dishes attached to towers. In this case, each communication link includes two dishes directed at each other. This method can still carry a large volume of calls, but it requires far less investment in terms of infrastructure. In general, the communications network between towers is structured in a hierarchical fashion originating from the local switching station. There are four basic requirements that must be in place for a user to make a successful cellular call. First, the user must be within the coverage area of a tower. Second, the tower must have sufficient available bandwidth to accept the call. Third, there must be residual capacity along the route between towers and switching stations to the recipient of the call. Finally, the recipient must be able to accept the call (i.e., be within coverage area that has a tower with capacity remaining or be on a connected land-line with available bandwidth). If a tower loses a microwave dish, or if a microwave dish is misaligned, then it will not be able to use that link to communicate with another tower. Similarly, a cut to a fiber-optic link or damage to the communication equipment might result in a loss of connectivity. A tower that cannot communicate with other towers or a switching station will be unable to accept calls, regardless of whether its users can communicate with the tower. The same holds true if a tower's links are operating at capacity, in which case no new users can place calls unless another user hangs up. Lastly, if a tower collapses or loses all power, then it cannot handle any calls, nor can it forward calls to other towers. Survivability issues of mobile wireless networks have been the subject of several studies (see e.g., [10], [11], and references therein). The important performance metrics of survivable cellular networks include call blocking probability, lost user load, forced call termination probability, and call setup delay. The common approaches for modeling the performance of cellular network include development of a simulation model that accurately captures a realistic failure scenario. Such models typically assume that the calls arrive to the system according to a Poisson process and the calls have exponentially distributed holding times.

2. Structural Reliability Modeling

One of the critical aspects of estimating the reliability of a cellular communications network during a hurricane is estimating the reliability of the structures (towers) that support the cellular antenna. These structures can be prone to failure during high wind events such as hurricanes. Structural reliability analysis [12, 13, 14] provides a natural approach for estimating the failure probability as a function of the wind load on a cellular tower. In general, both the wind load and resistance of a tower is uncertain. Letting $f_L(l)$ be the PDF for the wind load experienced at a given cellular tower and $F_R(r)$ be the CDF for the resistance of that cellular tower, the probability of failure for wind load w is given by:

$$p(f|w) = \int_x f_L(x)F_R(x)dx \tag{2.1}$$

The resistance of the structure is estimated using appropriate structural analysis methods such as finite element analysis. There are a number of approaches available for approximating the results of Equation 2.1 once the distributions of the load and resistance are estimated, including analytical approximations such as first and second order second moment methods as well as simulation-based approaches. The results of this type of analysis are generally presented as a fragility curve that shows the probability of failure of the structure as a function of the applied load.

For the purposes of demonstrating the integrated approach for estimating the reliability of a cellular network during a hurricane, we will use the fragility curves shown in Figure 1. These fragility curves are at the level of individual failure modes (e.g., foundation failure) rather than at the level of a complete cellular tower site

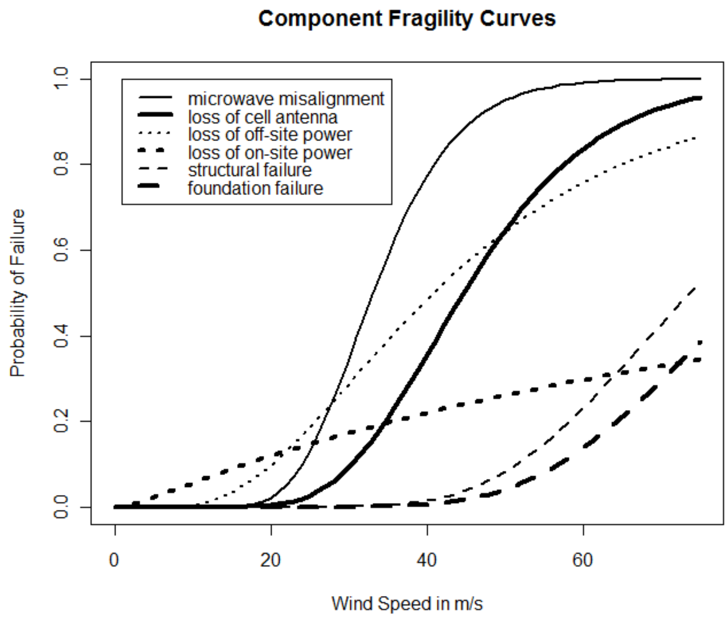


Fig. 1. Example Fragility Curves Used in Case Study.

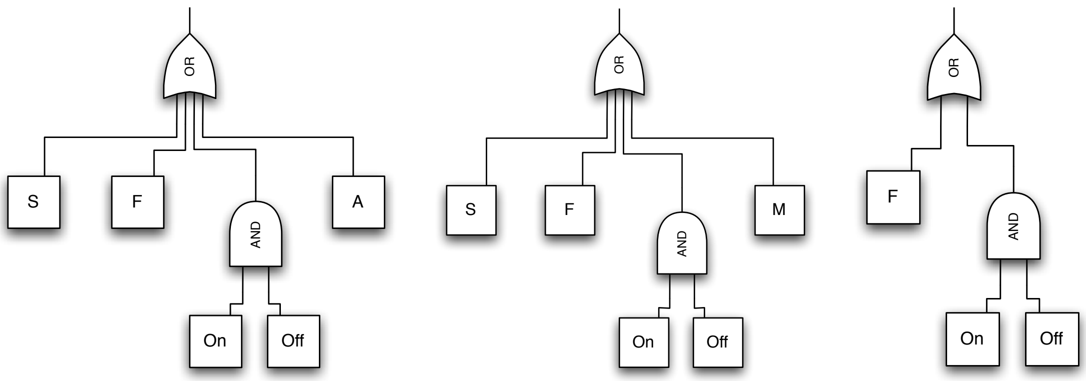
because, as discussed above, different failure modes have different implications for the functionality of the cellular site. For example, loss of cellular antenna at a given tower would prevent that tower from receiving or broadcasting calls, but it would still be able to relay calls from other towers if its microwave dish or fiber-optic link were functional. However, if the tower experiences structural failures, it would not be able to service incoming calls, outgoing calls, or relay calls through its microwave dish, but it would still be able to serve as a relay through its fiber-optic cable if the necessary power supply and supporting equipment were functional. The fragility curves in Figure 1 were not derived from analysis but rather based on engineering judgment, and they are for the purposes of demonstrating the approach only.

3. Hurricane Wind Field Modeling

Once the fragility curves have been defined for each failure mode of the cellular sites, the input loading is needed. For a hurricane, this requires the estimates of the wind speeds that would be experienced at each cellular site. We used a three-second gust wind speed, a common measure of hurricane wind intensity, as our measure of wind speed at a given cellular site. For the purposes of this case study, we based our wind field estimates on wind speed data estimated for Hurricane Ivan as it made landfall. The wind speed at each cellular site was estimated based on the hurricane wind field model developed by Huang et al. [15], and is the same model that was used in studies of power outages during hurricanes in both the Carolinas [16] and the Gulf Coast region [17]. In this hurricane wind field model, reconnaissance flight data is used to develop a gradient-level wind estimation model based on Georgiou's wind field model [18] and the hurricane decay model of Vickery and Twisdale [19]. This model produces an estimate of the gradient-level wind speed throughout the duration of a hurricane at the center of each cellular service area. This estimated wind speed was then converted to a "surface wind speed", the wind speed estimated at a height of 10 m in an assumed open exposure location, by using a multiplicative gradient-to-surface conversion factor. The gradient-to-surface conversion factor was taken to be 0.72 for sites more than 10 km from the coast, 0.80 for sites within 10 km from the coast, and 0.90 for sites adjacent to the sea as suggested by Rosowsky et al. [20]. We did not attempt to use different conversion factors based on records of local land cover types. We also did not correct for local topography effects because we did not have enough detailed information to include this in the model.

4. Reliability and Simulation Modeling

The combination of failure-mode level fragility curves and hurricane wind estimates provides the basis for estimating likelihoods of different types of losses of service for each cellular site within a given area. However, the failure-model level information must still be combined into site-level estimates of cellular site performance. Probabilistic Risk Analysis (PRA) provides a natural approach for this estimation. There exist three primary types of failures at each cellular site in a cellular communication network during hazardous wind loadings such as hurricanes. These include: (1) the inability of cellular towers to handle phone calls due to lack of network availability, (2) microwave dish failure (or misalignment), and (3) damage to fiber-optic links. Among these types of failures there exist six physical system states that contribute to each mode failure. These include, but are not limited to: (1) structural cellular tower collapse, (2) failure of tower foundation, (3) loss of onsite and offsite power, (4) loss of microwave dish, and (5) loss of cellular antenna. Figure 2 provides the fault tree for each of the three failure types, and shows how their system states lead to each of these types of failure. Figure 2(a) is the fault tree for cellular phone call failure, meaning that a cellular call cannot be handled by a cellular site. Figure 2(b) shows the fault tree for microwave dish failure. If the microwave dish fails, the cellular site cannot relay calls to other towers through its microwave dish. Finally, Figure 2(c) shows the fault tree for fiber-optic link failure. A fiber-optic link failure would prevent a cellular site from relaying calls via its fiber-optic cable. In all three fault trees, "s" is the event failure of the cellular tower structure itself, "f" is the event failure of the cellular tower foundation, "on" is the event loss of onsite power, "off" is the event loss of offsite power, "a" is the event loss of the cellular antenna at the site, and "m" is the event loss of microwave dish.



(a) Cellular Failure (b) Microwave Failure (c) Fiber Failure

Fig. 2. Fault Tree for Failures.

The fault trees in Figure 2 were developed based on an expert assessment of the failure modes for each of the three main types of failure of a cellular tower site. Note that the only redundancy that exists is in the power supply. We have assumed that off-site power and on-site power are in parallel and that their availability is not time-dependent. It is also likely that in many cases, off-site power would be lost during a hurricane [16, 17, 21]. Assuming that on-site power is not time-dependent ignores the reality that fuel stored on-site for on-site power are limited, and refueling cellular sites may prove prohibitively difficult in some locations after a hurricane in which there is heavy damage to transportation infrastructure. The on-site power supply, when functional, generally has enough fuel for several hours of operation. In this chapter we focus on the hours immediately after a hurricane in which cellular call traffic is generally both heaviest and most urgent as individuals and organizations respond to the hurricane. Beyond this initial period, additional analysis of the duration that backup power can be maintained would be needed, including the size of the fuel supply and the availability of the site for refueling.

5. Integrated Modeling Approach

The methods discussed above – cellular network performance modeling, structural reliability analysis, hurricane wind field modeling, and probabilistic risk analysis – provide the basis for an integrated approach for modeling the reliability of cellular network performance during future hurricanes. The process consists of the following steps:

1. Simulate a hurricane wind field corresponding to characteristics of the hurricane of concern using the wind field simulation models introduced above.
2. Estimate the probability of each failure mode for each of the three types of failures occurring at each cellular site based on the simulated wind speed.
3. Estimate the probability of a cellular site experiencing each of the three types of failure – inability to handle calls, inability to relay calls via microwave dish, and inability to relay calls via fiber-optic cables – based on the PRA discussed above.
4. Simulate a system state for a set of failure types experienced at each cellular site in the system through Monte Carlo simulation based on the PRA results.
5. For each simulated system state, simulate the performance of the cellular system of the region of interest over time after a hurricane makes landfall.

If multiple hurricane scenarios such as randomly generated storms are required, then the process above can be repeated for each of the simulated storms.

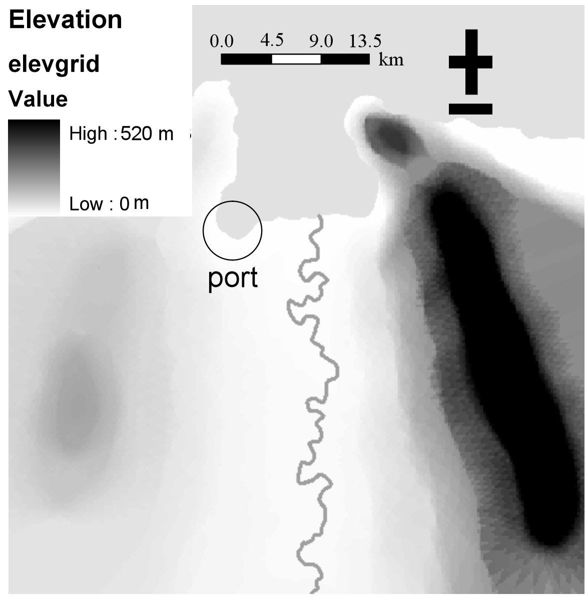


Fig. 3. Mesopolis Topography.

C. Case Study Network

We used the synthetic city Mesopolis as our case study for demonstrating the integrated modeling approach [22]. Mesopolis is a synthetic city representing a coastal city with a population of approximately 125,000. It was developed at Texas A&M University to serve as a test bed for developing methods used for assessing infrastructure performance. Our choice of a synthetic city was driven by the lack of availability of real cellular topologies. We extended the basic Mesopolis testbed by designing a cellular communication system based on a standard pattern of hexagonal cellular coverage. As shown in Figure 3, the Mesopolis topographical characteristics include two longitudinal ridges that extend into a double peninsula at the northern face. A valley exists between the two ridges, with a river flowing along the valley center and into the ocean, forming a delta at the outlet.

Because cellular networks are typically laid out in hexagonal patterns, a cellular

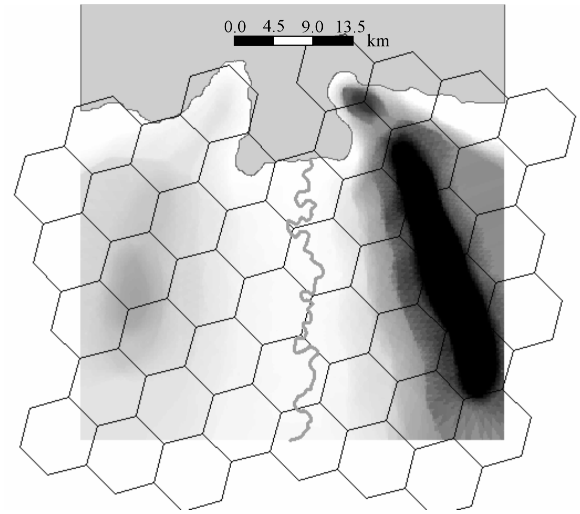
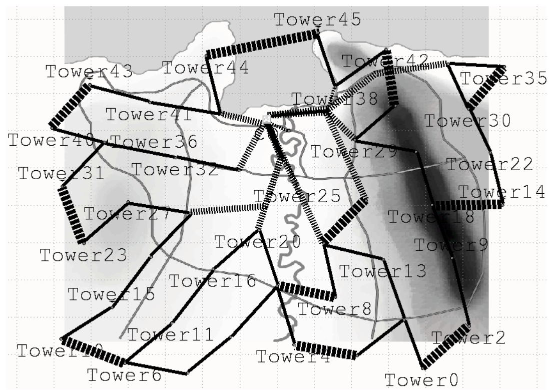


Fig. 4. Mesopolis Cellular Coverage.

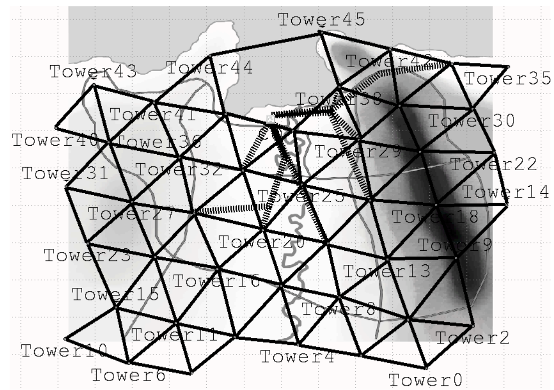
system based on a hexagonal grid was developed for Mesopolis. Each hexagon was given a radius of 6.44 km (4 mi), typical of current cellular systems. The grid was then rotated and moved to place a large series of towers on top of the ridges, as well as providing coverage for the bay. The resulting cellular grid is shown in Figure 4.

Mesopolis does have an assumed development timeline that influenced how the different portions of the city were designed, and we designed the cellular network to be sensitive to this. For example, the cellular switching station was placed in the older part of the city at the location of the telephone company. Towers that are located in this same area are connected with fiber-optic connections to the switching station. Nearby towers which are not connected via fiber-optic are connected with microwave dishes to the nearest connected tower. This creates a tree of connections originating at the switching station and moving out to connect all towers in Mesopolis. This configuration is illustrated in Figure 5(a).

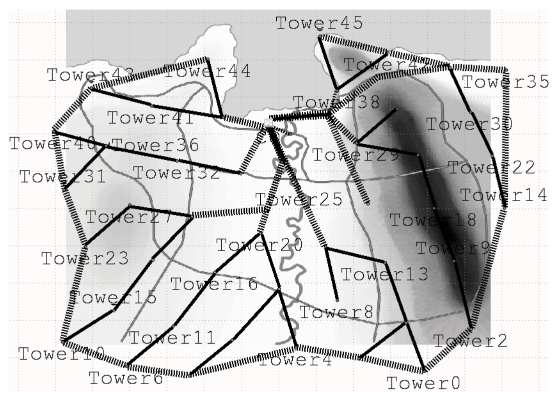
For the given communication system design for Mesopolis, if a single connection is lost, one or more towers will be out of service. While this may be realistic, it is not



(a) Tree Layout



(b) Mesh Layout



(c) Outer Optical Ring

Fig. 5. Test Layouts of the Mesopolis Cellular System.

resilient to failure. In order to study methods for improving the network's reliability, additional microwave links were added on the outside of the network, denoted by the hashed lines. This made the network double-connected, meaning it required a minimum of two connection failures in order for a tower to lose communications. In addition to studying the effect of adding minimal redundancy, we also examined different network 'design' scenarios. First, we tested the original and redundant networks with unlimited bandwidth on each link to remove the effect bandwidth on the overall system. Second, we artificially made each microwave link in the original network invulnerable to the effects of wind so we could see what effect the microwave

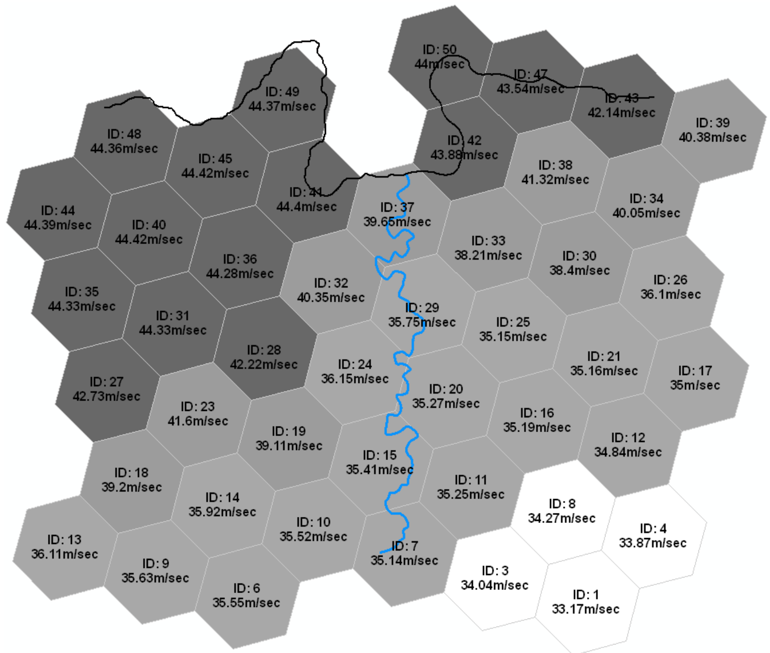


Fig. 6. Hurricane Wind Field Used in Case Study. Darker Hexagons Indicate Stronger Wind Fields.

links had on the system. Third, we tested alternative topologies by extending the original network to a full mesh network by adding microwave links from every tower to each of its neighbors as illustrated in Figure 5(b). Finally, we tested an optical network connecting towers on opposite sides of the coast through the towers on outside of Mesopolis as shown in Figure 5(c).

For the purposes of our case study, we superimposed the hurricane wind field estimates over Mesopolis based on estimated wind data from Hurricane Ivan. We did this by rotating the wind field 180 degrees so that the hurricane approached the city from the north and then estimated the maximum three-second wind speed at the centroid of each cellular hexagon. The resulting wind estimates are shown in Figure 6.

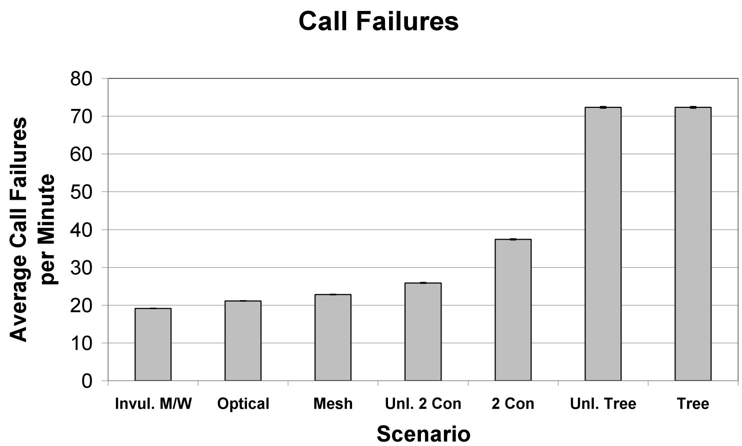


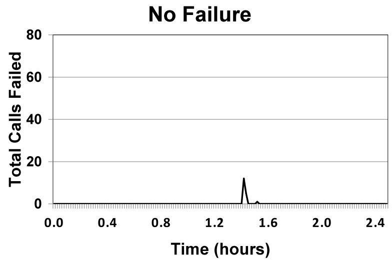
Fig. 7. Average Failure Rates for All Scenarios.

D. Results

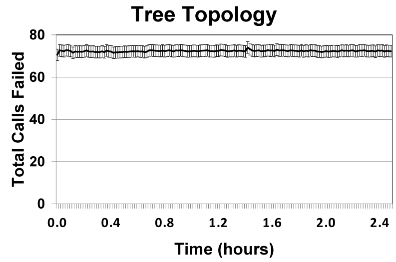
We simulated the behavior of the cellular network across a series of failures. Each tower may lose its cellular antennae, microwave dishes, fiber-optic links, or any combination therein. The set of these failures across all towers defines a network's state. By utilizing the fragility curves in Figure 1 and the wind speed model in Figure 6, we calculated estimates of failure event probabilities for each tower. Finally, from the fault trees in Figure 2, we defined the state of the network. Running this simulation 5,000 times gave us 3,383 unique states for the network.

The cellular network for Mesopolis was modeled using the OPNET Modeler®[23], and each unique state was simulated for a 2.5 hour duration. We simulated the same states in all our scenarios to compare the results. We collected statistics for each network element, including how many calls were made per minute and how many failed per minute. A call may fail if network elements are broken, such that there is no route from the caller to the destination. In addition, a call may also fail if some element on the path to a destination is too busy to accept another call. The graph in Figure 7 shows the average call failure rate in each network topology. The results suggest that the original tree structure with invulnerable microwave links is the network with the lowest call failure rate but that the optical outer ring network and the microwave mesh network perform nearly as well. The original tree network, the network representing a realistic cellular network system, performs the worst of the considered systems by a large margin, and even if the bandwidth limitations on this network are removed its performance does not increase appreciably. When judged based on the average call failures per minute, the extra redundancy offered by the optical outer ring and mesh networks offers a significant increase in system availability in a post-hurricane situation. The graph in Figure 8 shows each topology's average call failure rate over time. It is important to note that some calls may fail due to a momentary network overload without any failure, as can be seen in the no damage case in Figure 8(a) which has a momentary peak at 1.4 hours. The results in Figure 8 reflect those in Figure 7 in that for all topologies, the network quickly achieves it steady-state performance in terms of the number of calls failing per minute. The one exception to this is for the double-connected tree topology where there is an initial increase in the number of failed calls per minute. This is due to redundant links providing a longer backup bath than the primary path. As future calls are made, the longer paths being used as backup paths become capacity-limited, leading to more dropped calls.

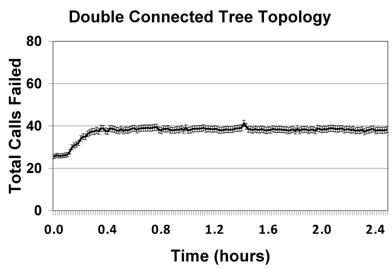
We also analyzed the percentage of call failures by tower as shown in Figure 9 where we can see which towers experienced the highest failure rate. Figure 9 shows that the towers along the coast experienced the highest failure rate, which is expected because they would have suffered the highest winds. Towers with the longest route



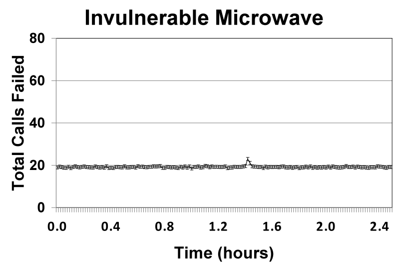
(a) No Failure



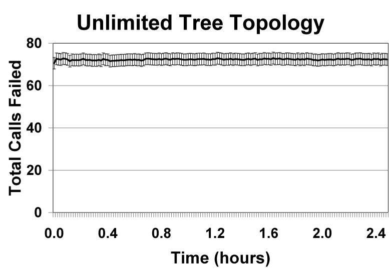
(b) Tree Topology



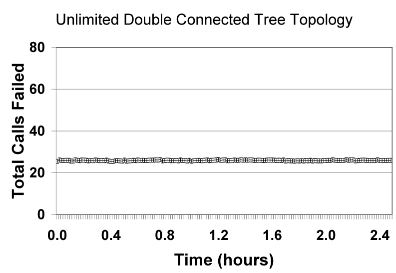
(c) Double Connected Tree Topology



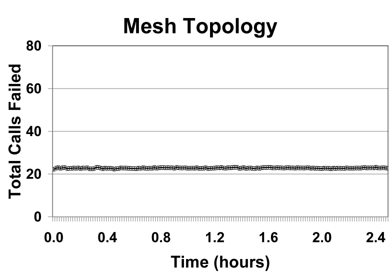
(d) Invulnerable Microwave



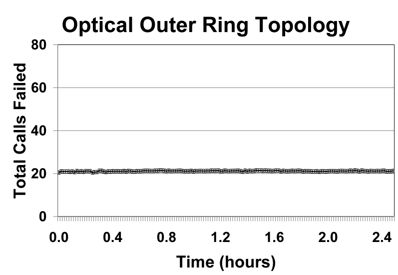
(e) Unlimited Tree Topology



(f) Unlimited Double Connected Tree Topology



(g) Mesh Topology

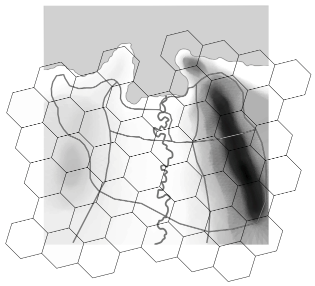


(h) Optical Outer Ring

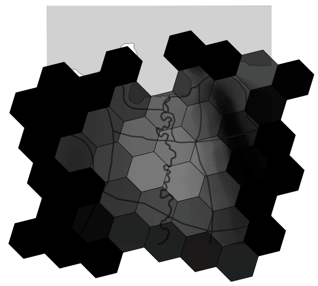
Fig. 8. Average Call Failure Rates.

to the central office (e.g., those on the outer edges of the city) also experienced a higher failure rate, since any link failure along this path could result in its inability to contact other towers.

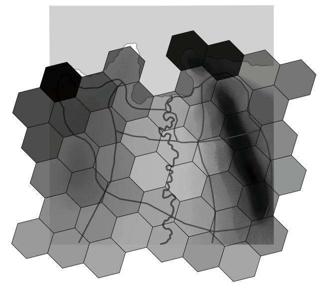
The double-connected network greatly mitigated the effect of long chains of links where any one failure could result in the failure of the last tower. Figures 7 and 8 show that these additional links decrease the call failure rate when compared to the tree topology, but in the first half hour the failure rate increases while longer call paths saturate the bandwidth on links. In order to determine how well the tree topology could be improved, we tested it while not allowing any microwave links to fail. Since failed microwave links are responsible for most of the failed calls in the tree topology, this network is expected to perform the best. Figures 7 and 8 confirmed this expectation with this new network yielding the lowest call failure rate of the tested networks. When we analyzed the tree topology and double-connected topology where all links had unlimited bandwidth, we noticed that the tree topology saw very little benefit from the increased bandwidth. This is because a vast majority of its failures are due to a loss of connectivity, not lack of bandwidth. The double-connected network saw benefit from the increased bandwidth because when failure occurs, some towers will route calls through different towers than when a failure does not occur. In this network, some of the call failures result from a tower sharing its bandwidth with additional towers, increasing the demand on remaining links. This can be seen in Figure 8 where the call failure rates in the double-connected topology increase over time until network links are saturated, but the rate does not increase when the bandwidth is not limited. The mesh topology improved the network's performance over the redundant links in the double-connected network as shown in Figures 7 and 8. This is expected both due to the increase in redundant links as well



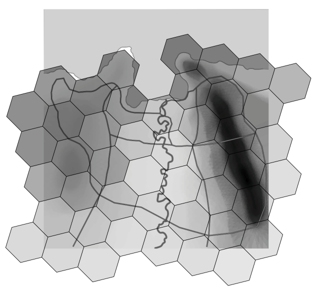
(a) No Failure



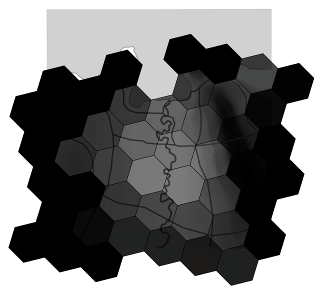
(b) Tree Topology



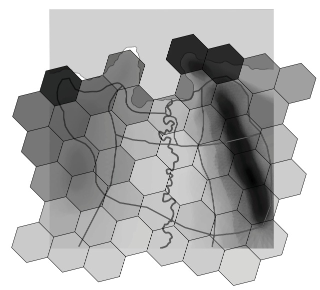
(c) Double Connected Tree Topology



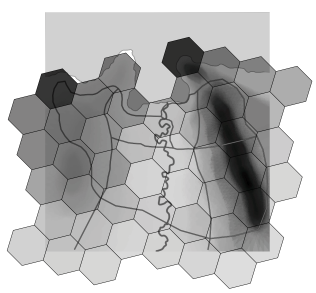
(d) Invulnerable Microwave



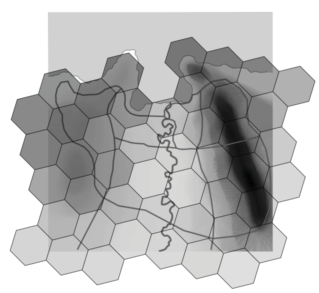
(e) Unlimited Tree Topology



(f) Unlimited Double Connected Tree Topology



(g) Mesh Topology



(h) Optical Outer Ring

Fig. 9. Calls Failed Per Tower; Black Represents a 10% Failure Rate.

as the increase in bandwidth. The optical outer ring improved performance more than other topology changes. Since the optical link is more reliable than microwave links and carries more bandwidth, this is expected to have the greatest improvement. This topology achieved a failure rate within 10% of the scenario where microwave links were not allowed to fail.

E. Discussion and Conclusions

Our results show that the reliability of cellular networks during hurricanes can be evaluated using an interdisciplinary approach that combines structural reliability analysis, probabilistic risk analysis, network traffic modeling, and simulation. The methods provide a basis for both assessing the reliability of an existing network in a computationally feasible manner and for assessing improvements in network performance associated with changes in network topology. Our results also suggest that increasing path redundancy in some form can significantly increase the reliability of cellular networks during hurricanes. A microwave neighbor-to-neighbor mesh configuration and the addition of an optical fiber ring around the perimeter of the network are particularly effective forms of providing redundancy for our case study network. The results of this work should provide a basis for assessing the reliability of existing cellular networks in hurricane prone areas as well providing a basis for assessing proposal to improve these systems through topological modifications.

CHAPTER III

EFFICIENT TRAFFIC LOSS EVALUATION FOR TRANSPORT BACKBONE NETWORKS

A. Introduction

Communication networks play an important role in many social and economic activities. Interruptions in data transmission and exchange, even for a short period of time, may suspend critical operations and lead to a significant loss of revenue. In this chapter we focus on the resilience and survivability of fiber optic networks which serve as a backbone of the modern communication infrastructure.

Due to their ubiquitous deployment, optical networks are prone to failures. While a considerable effort has been devoted to improving the physical protection of underground and underwater cables, fiber cuts occur at a significant rate. Since each optical link has a very high capacity (up to several terabits per second), and usually transmits data from multiple connections, a link failure may result in a significant loss of traffic. As a consequence, the survivability of optical networks has become an important research direction with numerous protection techniques proposed and implemented over the last decade. In particular, there are two types of dedicated protection which are widely used in practice: 1+1 protection and $1:1$ protection [4]. In 1+1 protection, each connection is allocated two link-disjoint paths such that the data is transmitted on both paths from the source to destination. In the case of a failure of a link in the primary path, the destination node switches over to the secondary path. In $1:1$ protection schemes, the backup path is not used unless the primary path fails.

While $1+1$ and $1:1$ protection techniques can handle single edge failures, some traffic may be lost in the case of multiple link failures. As demonstrated by connection losses in Egypt in early 2008, simultaneous link failures, while rare, can still occur at a non-negligible rate. Thus, assessing the reliability and survivability of optical networks is of major importance for service providers. Accordingly, in this chapter we present efficient tools and techniques for computing the expected loss of traffic (ELT) in optical networks with 1+1 protection. Our study employs both traditional methods based on Monte Carlo simulations as well as analytical methods based on cut set enumeration. To facilitate the computational process, we also employ artificial intelligence methods based on genetic algorithms. Our simulation results show that the proposed methods accurately estimate traffic loss in practical settings.

B. Background and Prior Work

Computing the failure probability of a network is a classical problem in the reliability theory (see e.g., [24, 25, 26, 27, 28, 29, 30, 31, 32, 33] and references therein). The traditional methods rely on the Monte-Carlo simulations [28]. A randomized fully-polynomial time approximation scheme (FPRAS) for the all-terminal network reliability problem has been presented in [34].[1] Karp and Luby [25] presented a general framework for the construction of Monte Carlo algorithms for estimating the failure probability of a multiterminal planar network. Liu and Trivedi [29] proposed several measures of network survivability. Samaan and Singh [30, 31, 32, 33] proposed genetic algorithms for analyzing the reliability of power networks.

Analyzing availability and reliability of optical networks has also received a significant attention from the research community [35, 36, 37, 38, 39, 40]. Jereb [36] presented an overview of the main directions of network reliability analysis. Jereb

[1] An FPRAS is a randomized algorithm that given a parameter ε finds, with high probability, an ε-approximate solution in time that is polynomial in the input size and $1/\varepsilon$. A ε-approximate solution is a solution which is accurate within a relative error of ε.

et. al. [35] investigated the effectiveness of stratified sampling in WDM networks. Levendovszky et. al. [37] introduced several statistical algorithms for network reliability analysis based on adaptive approximation and deterministic radial basic function (RBF) method. A heuristic search algorithm inspired by evolutionary methods for all terminal reliability design problem has been proposed in [38]. Clouqueur and Grover [41] have presented methods for the availability analysis of several restoration mechanisms for mesh networks.

C. Model

We model the optical network, N, by a connected undirected graph, $G(V, E)$. The set of nodes, V of G, represents the routers and switches, while the set of edges, E of G, represents the communication links. For each edge, $e_i \in E$, we denote $c(e_i)$ as the capacity of edge e_i, i.e., the maximum amount of traffic flow that can be routed through e_i. A cut, $C \subseteq E$, in graph $G(V, E)$ is a set of edges whose removal results in a partition of G into at least two components. We say that a cut C separates nodes v and u if every path from v to u includes at least one edge in C.

For each edge, $e_i \in E$, we also associate a certain probability of failure, $p(e_i)$. The probability of failure may depend on several factors, such as link location, the length of the optical fiber, geographic area, etc. We assume that all edge failures are independent and all nodes in the network are perfectly reliable. We also assume that the probabilities of link failure are given and we use them as input to our algorithms. Methods for determining the reliability of optical cables are discussed in several previous studies (see e.g., [42, 43]).

In our model, the network supports a connection between each pair of distinct nodes in V. We use Z as the set of such connections and $M(v, u)$ as the amount of traffic generated by the connection $(v, u) \in Z$. For clarity, we assume $M(v, u) > 0$ for each pair (v, u).

1. Network Reliability and Traffic Loss

A network state is captured by a vector, $x = (x_1, \cdots, x_m)$, such that $x_i = 0$ if edge e_i is operational, while $x_i = 1$ if edge e_i is faulty. We use X as the set of all possible system states. We also use $F(x) \subseteq E$ as the set of faulty edges in state $x \in X$. For each state $x \in X$ we use $p(x)$ as the probability of x.

The network reliability $R(N)$ is defined as follows:

$$R(N) = \sum_{x \in X} p(x) \cdot I(x), \tag{3.1}$$

where $I(x) \in \{0, 1\}$ is the indicator function, such that $I(x) = 0$ if the network is disconnected in state x and $I(x) = 1$ if it is connected. The network is said to be disconnected in state x if there are two nodes, v and u, such that v is not reachable from u at this state.

Let $L_{(v,u)}(x)$ be the amount of traffic lost by connection (v, u) at state x. The total amount of traffic $L(x)$ which cannot be routed at state x is defined as:

$$L(x) = \sum_{(v,u) \in Z} L_{(v,u)}(x). \tag{3.2}$$

The expected loss of traffic $L(N)$ of the network N is defined as follows:

$$L(N) = \sum_{x \in X} p(x) \cdot L(x). \tag{3.3}$$

The traffic loss model described above assumes that all connections have identical availability requests. The model can be extended to support availability-differentiated connections by introducing a cost, $c_{(v,u)}$, of losing a unit of flow for each connection,

$(v,u) \in Z$. We then will use a weighted traffic loss $\bar{L}(x)$ instead of $L(x)$ for computing the value of $L(N)$, where $\bar{L}(x)$ is defined as follows:

$$\bar{L}(x) = \sum_{(v,u)\in Z} c_{(v,u)} \cdot L_{(v,u)}(x). \tag{3.4}$$

Since the goal of the restoration strategy is to minimize the values of $L(x)$ for each state x, it will favor the connections with high values of $c_{(v,u)}$. In this chapter, we focus on settings with identical availability requests, i.e., $c_{(v,u)} = 1$ for all $(v,u) \in Z$.

The main problems considered in this chapter can be formulated as follows.

Problem NR (Network Reliability) *Given a communication network, N, represented by a graph, $G(V,E)$, with failure probability, $p(e_i)$ specified for each edge, $e_i \in E$, compute the value of $R(N)$.*

Problem ELT (Expected Loss of Traffic) *Given a communication network, $G(V,E)$, represented by a graph, $G(V,E)$, edge failure probabilities, $p(e_i)$, capacities, $c(e_i)$, a routing strategy, and a protection strategy, determine the expected loss of traffic $L(N)$.*

Problem NR has been the subject of many studies in the reliability theory (see e.g., [34] and references therein). The problem belongs to the complexity class $\#P$ which is associated with counting solutions to problems in NP. It was shown that this problem is $\#P$ complete; that is, this problem is as hard as any other problem in this class. Since the complexity class $\#P$ is at least as intractable as NP, it is very unlikely that there exist any polynomial time solutions. However, Problem NR admits an FPRAS, i.e., it is possible to estimate the value of $R(N)$ to within a relative error of $1 \pm \varepsilon$ with high probability in polynomial time with respect to $|V|$ and $1/\varepsilon$ [34]. Note that computing the expected loss of traffic is a more general problem than Problem NR.

2. Routing and Protection Strategies

In general, the methods presented in this chapter can be used with any routing algorithm. In our test case, we use $1+1$ protection scheme and employ the following routing strategy: For each connection, $(v, u) \in Z$, we find two edge-disjoint paths, $P^1_{(v,u)}$ and $P^2_{(v,u)}$, that have the minimum total number of hops. The paths are found by using the algorithm from Suurballe and Tarjan [44]. Next, we reserve $M(v, u)$ units of bandwidth on all edges that belong to $P^1_{(v,u)}$ and $P^2_{(v,u)}$. We assume that network is over-provisioned by the factor of two, that is, for each edge, $e_i \in E$, the capacity, $c(e_i)$, of e_i is at least two times the total amount of bandwidth reserved on e. Over-provisioning is a commonly used technique for improving network reliability and robustness (see e.g., [40]).

For each faulty state, $x \in X$, we restore all connections, $(u, v) \in Z$, for which both paths $P^1_{(v,u)}$ and $P^2_{(v,u)}$ include failed edges. We denoted the set of such connections as S. All other connections will continue to send data through the one or two disjoint paths computed in the initial routing stage. We restore all failed connections in a sequential order. For each such connection, (v, u), we find a shortest path between v and u that has residual capacity of at least $M(v, u)$. In this case, $L(x)$ includes the total amount of traffic which cannot be rerouted. If the restoration strategy supports availability differentiation, it will first restore connections (v, u) that have high cost $c_{(v,u)}$ of loosing traffic.

Integer Programming Approach

A better solution can be obtained by using the integer programming approach. Let $S \subseteq Z$ be a set of connections, (v, u), for which both paths $P^1_{(v,u)}$ and $P^2_{(v,u)}$ include failed edges. For each edge, $e_i \in E$, we denote $r(e_i)$ as the residual capacity of e_i, i.e., the difference between the capacity $c(e_i)$ of e_i and the total bandwidth reservations made by connections that do not belong to S. We also denote ρ as the portion of the required traffic flow of each connection in S that can be rerouted. The following integer linear program determines the maximum value of ρ.

$$\begin{array}{ll}
\text{Maximize } \rho & \\
\text{Subject to:} & \\
\displaystyle\sum_{e\in(w,u)} u^e_{(v,u)} = 1 & \text{for } (v,u)\in S \\
\displaystyle\sum_{e\in(j,w)} u^e_{(v,u)} - \sum_{e\in(w,j)} u^e_{(v,u)} = 0 & \text{for } (v,u)\in S \\
 & \text{and } w\in V\setminus\{v,u\} \\
\displaystyle\sum_{(v,u)\in S} u^e_{(v,u)}\cdot\rho\cdot M(v,u) \le r(e) & \text{for } e\in E \\
u^e_{(v,u)}\in\{0,1\} & \text{for } (v,u)\in S \text{ and } e\in E
\end{array}$$

Here, $u^e_{(v,u)}$ is the indicator variable that specifies whether the connection (u,v) will be rerouted through edge e. The maximum value ρ^* of ρ can be found through binary search that evaluates, at each step, the feasibility of the integer linear constraints by using Integer Programming. With this approach, the loss of traffic at state x is equal to $(1-\rho^*)\sum_{(v,u)\in Z} M(v,u)$ if $\rho^* < 1$ and zero otherwise.

D. Cut Enumeration

The standard methods for estimating network reliability are built around Monte Carlo sampling methods. In Section G we show that, under certain conditions, Monte Carlo methods can be efficiently used for Problem ELT as well. However, the Monte Carlo sampling methods suffer from relatively slow convergence rate due to a large variance of the corresponding estimator. In particular, if the probabilities of edge failures are small, the vast majority of randomly picked assignments will produce states with no traffic loss, resulting in high computational time.

The basic idea is to enumerate high-probability cuts that disconnect the network into two components. As shown in [45] the number of such cuts is relatively small. Minimum two-way cuts can be efficiently enumerated by using the algorithm presented in [46]. This algorithm can be modified to produce cuts with decreasing probability of failure.

Let $\mathbb{C} = \{C_1, \ldots, C_k\}$ be the set of k cuts with highest probability of failure. We say that a cut $C_i \in \mathbb{C}$ is failed in state x if $C_i \subseteq F(x)$. Clearly, a failure of a cut results in a loss of traffic since no rerouting strategy can reroute traffic when no route exists. The goal of our algorithm is to compute the expected loss of traffic $\hat{L}(N)$ due to the failure of cuts in $\mathbb{C}$.

To define $\hat{L}(N)$ formally, we need to introduce the following definitions for each state $x \in X$:

1. $\mathbb{C}(x) \subseteq \mathbb{C}$ - set of failed cuts in state x, i.e.,

$$\mathbb{C}(x) = \{C_i \in \mathbb{C} \mid C_i \subseteq F(x)\};$$

2. $Z(x) \subset Z$ - set of connections $\{(v, u)\}$ disconnected by cuts in $\mathbb{C}(x)$, i.e.,

$$Z(x) = \{(v, u) \mid \exists C_i \in \mathbb{C}(x) : C_i \text{ separates } v \text{ and } u\};$$

3. $\hat{L}(x)$ - the amount of traffic lost due to failed cuts in $\mathbb{C}(x)$, i.e.,

$$\hat{L}(x) = \sum_{(v,u) \in Z(x)} M(v, u).$$

The value of k is selected in such that the total contribution of other cuts to the expected traffic loss is negligible. Our experimental results show that accurate estimation can be achieved with as little as 1000 cuts. Note that if there are no upper bounds on edge capacities or if the network is sufficiently over-provisioned, then

$$\hat{L}(N) = \sum_{x \in X} p(x) \cdot \hat{L}(x)$$

provides a good estimate of $L(N)$. In fact, Our numerical results show a 2% difference between $\hat{L}(N)$ and $L(N)$ when the network is over-provisioned by the factor of 4.

As mentioned above, our goal is to accurately estimate the value of $\hat{L}(N)$. Note that the total traffic loss due to a failure of a cut $C_i \in \mathbb{C}$ can be efficiently computed; however, counting expected traffic loss due to all cuts in $\mathbb{C}$ poses challenges because of the correlations between them. Accordingly, we use the DNF counting techniques described in [45] for accurate estimation of $\hat{L}(N)$.

The formal description of Algorithm Cut Enumeration is presented in Figure 10. The algorithm consists of two phases. The first phase (Steps 1-5 of the algorithm) computes, for each cut $C_i \in \mathbb{C}$, the expected traffic loss T_i due to a failure of cut C_i. This is done by first identifying the set of connections Z_i whose endpoints are separated by C_i and then computing the total traffic generated by these connections. Parameter T computed in Step 5 is an upper bound on $\hat{L}(N)$ because the dependencies of cuts in $\mathbb{C}$ cause some traffic to be counted multiple times. The purpose of the second phase (Steps 6-13 of the algorithm) is to compute the value of random variable Y, which is an unbiased estimator of $\hat{L}(N)$ with small variance. The variance of Y depends on the number of iterations, n, which is given to the algorithm as input. The rest of this section is devoted to establishing the correctness of the algorithm.

The following theorem shows that random variable Y is an unbiased estimator for $\hat{L}(N)$.

Algorithm Cut Enumeration (n)

1 **For** $i \leftarrow 1$ to k **do**
2 $\quad Z_i \leftarrow \{(v,u) \mid C_i$ separates v and $u\}$;
3 $\quad M(Z_i) \leftarrow \sum_{(v,u)\in Z_i} M(v,u)$
4 $\quad$ Compute the expected traffic loss T_i from connections due to a failure of cut C_i:
$\quad T_i \leftarrow M(Z_i) \cdot \prod_{e\in C_i} p_e$.
5 $T \leftarrow \sum\limits_{C_i\in\mathbb{C}} T_i$
6 **For** $i \leftarrow 1$ to n **do**
7 $\quad$ Pick a cut $C_j \in \mathbb{C}$ with probability $\frac{T_j}{T}$.
8 $\quad$ Pick a connection $(v,u) \in Z_j$ with probability $\frac{M(v,u)}{M(Z_j)}$
9 $\quad$ Pick a failure state x in which all edges that belong to C_j are faulty and the state of all other edges is chosen at random according to their probability of failure.
10 $\quad$ Find all cuts $C_l \in \mathbb{C}$ that satisfy the following two conditions:
$\quad$ (a) $C_l \subseteq F(x)$
$\quad$ (b) C_l separates v and u
11 $\quad \tau_i \leftarrow$ the number of cuts found in Step 10
12 $\quad Y_i \leftarrow \frac{T}{\tau_i}$
13 $Y \leftarrow \dfrac{\sum_{i=1}^{n} Y_i}{n}$
14 **Return** Y

Fig. 10. Algorithm Cut Enumeration

Theorem 1 *Random variable Y is an unbiased estimator for $\hat{L}(N)$.*

Proof: It is sufficient to show that the value of Y_i, $i = 1, \cdots, n$, computed in Step 12 of the algorithm is an unbiased estimator of $\hat{L}(N)$.

First, we analyze the probability that the algorithm chooses a connection (v,u) and a state x in Steps 8 and 9. Steps 7-9 of the algorithm imply that this probability is equal to

$$\sum_{C_j:C_j\in F(x)} \frac{T_j}{T} \cdot \frac{M(v,u)}{M(Z_j)} \cdot \frac{p(x)}{\prod_{e\in C_j} p_e} = \sum_{C_i:C_j\in F(x)} \frac{M(v,u)\cdot p(x)}{T} = \\ = \frac{\tau_i \cdot M(v,u)\cdot p(x)}{T} \tag{3.5}$$

if $(v,u) \in Z(x)$ and zero otherwise. The first equality is obtained by substitution of T_j (see Step 4), and the second equality is obtained by noticing that all additive terms are equal and that the number of additive terms is equal to τ_i.

Next, we compute the expectation $E[Y_i]$ of Y_i as follows

$$\begin{aligned}E[Y_i] &= \sum_{x \in X} \sum_{(v,u) \in Z} \Pr[x \text{ and } (v,u) \text{ are picked}] \cdot Y_i = \\ &= \sum_{x \in X} \sum_{(v,u) \in Z(x)} p(x) \cdot M(v,u) = \sum_{x \in X} p(x) \sum_{(v,u) \in Z(x)} M(v,u) \\ &= \sum_{x \in X} p(x) \cdot \hat{L}(x) = \hat{L}(N).\end{aligned}$$

The first equality is a definition of the expected value of Y_i. The second equality is obtained by using the expression obtained in Equation (3.5). The third equality is obtained by taking the common factor $p(x)$ outsize the summation. The forth and fifth equalities follow from the definitions of $\hat{L}(x)$ and $\hat{L}(N)$, respectively.

■

Next, we show that the standard deviation $\sigma(Y_i)$ of Y_i for $i = 1, \cdots, n$ is bounded.

Lemma 2

$$\frac{\sigma(Y_i)}{E(Y_i)} \leq k.$$

Proof: Let $\alpha = \frac{T}{k}$. Note that $1 \leq \tau_i \leq k$. Since $Y_i = \frac{T}{\tau_i}$ (see Step 12) it holds that $\alpha \leq Y_i \leq \alpha \cdot k$, and, in turn, $E(Y_i) \geq \alpha$. This means that the random variable Y_i deviates from its mean by at most $k \cdot \alpha$. Hence, the standard deviation of Y_i is bounded by $k \cdot \alpha$. Since $E(Y_i) \geq \alpha$ it holds that $\frac{\sigma(Y_i)}{E(Y_i)} \leq k$. ■

We conclude that by sampling Y_i a sufficient number of times ($n = O(\frac{k^2}{\varepsilon^2})$) and outputting the mean, we can obtain an approximation of $\hat{L}(N)$ with high accuracy.

Lemma 3 *If the Algorithm* Cut Enumeration *is applied with* $n = O(\frac{k^2}{\varepsilon^2})$ *then it* *holds that*

$$Pr\left[|Y - \hat{L}(N)| \leq \varepsilon \cdot \hat{L}(N)\right] \geq \frac{3}{4},$$

Proof: The Chebyshev inequality implies that

$$Pr\left[|Y - \hat{L}(N)| \leq \varepsilon \cdot \hat{L}(N)\right] \geq \left(\frac{\sigma(Y)}{\varepsilon E[Y]}\right)^2.$$

Since $E[Y_i] = E[Y]$ and $\sigma(Y) = \frac{\sigma(Y_i)}{\sqrt{n}}$ it holds that

$$\left(\frac{\sigma(Y}{\varepsilon E[Y]}\right)^2 = \left(\frac{\sigma(Y_i)}{\varepsilon\sqrt{n}E[Y_i]}\right)^2.$$

By Lemma 2 implies that $\frac{\sigma(Y_i)}{E(Y_i)} \leq k$. Hence, for $n \geq \frac{4k^2}{\varepsilon^2}$ it holds that

$$\left(\frac{\sigma(Y)}{\varepsilon\sqrt{n}E[Y]}\right)^2 \geq \frac{3}{4}$$

and the lemma follows. ■

The probability $Pr[|Y - \hat{L}(N)| \leq \varepsilon \cdot \hat{L}(N)]$ can be increased by taking a larger number of samples.

E. Genetic Algorithms

The primary use of a *genetic algorithm* is to find an optimal or near-optimal solution for an optimization problem. A genetic algorithm is a simulation of evolution where the rule of survival of the fittest is applied to a population of individuals. In the basic genetic algorithm [47], an initial population is created randomly. Population individuals, called *chromosomes*, are then evaluated by applying some function or formula. A new population is selected from the old one based on the fitness value of the individuals. Genetic operators are then applied to the newly selected population to create the next generation. The most commonly used genetic operators are *crossover* and *mutation*. The process is repeated from one generation to another until

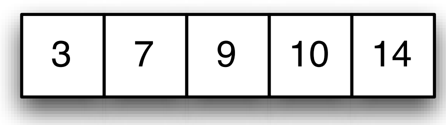


Fig. 11. Sample Chromosome.

a stopping criterion is reached.

In this study, we use a genetic algorithm to find states of the network that incur large amount of traffic loss. Since we are only assuming edge failures, we define each state $x \in X$ of the network as a variable-length array of faulty edges. Edges are sorted by increasing indexes and no repetition is allowed. The example shown in Figure 11 shows a chromosome that corresponds to a state in which edges 3, 7, 9, 10, and 14 have failed.

Each chromosome in our genetic algorithm represents a single state. The *fitness* $f(x)$ of each chromosome corresponding to state x is defined by the probability, $p(x)$, of x if x incurs a loss of traffic and 0 otherwise. This means that the most fit chromosomes correspond to the most likely states which still result in a loss of traffic.

Typically, a genetic algorithm is initialized with random values, and allowed to search for the most fit individual. In our model, we are not interested in the most fit individual, but rather the set of most fit individuals. Accordingly, instead of examining the final outcome of the algorithm, we look at all individuals examined by the algorithm across all generations. We use a hash table that records all individuals processed by the algorithm so far. Entries in the hash table are indexed by the state, and contain the amount of traffic loss and the probability of that state. The use of a hash table also allows us to accelerate the algorithm because when a state is encountered a second time, we can simply return the result we computed the in previous evaluation, eliminating the need to perform the flow calculations again.

To prevent the algorithm from converging on a single individual, we employ the *niching* technique on the chromosome space [48]. With niching, we penalize a chromosome's fitness if it is closer than a certain threshold, Q, to the individuals in the same generation. Specifically, let $\overline{X}$ be the individuals discovered by the algorithm in a particular generation. For any two individuals, x and y, we denote $d(x, y)$ as the Hamming distance between x and y. Then, the new fitness function $\hat{f}(x)$ is defined as follows:

$$\hat{f}(x) = \frac{f(x)}{\sum_{y \in \overline{X}}^{m} g(x, y)}, \tag{3.6}$$

where

$$g(x, y)) = \begin{cases} 1 - \left(\frac{d(x,y)}{Q}\right)^2 & \text{if } d(x, y) < Q \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

In addition to a custom representation for individuals, we also designed custom operations for *mutate* and *crossover*. We defined the mutate operation to randomly modify a single edge's index to another index. If the new index is already in the chromosome, the duplicate is simply removed. The mutation operator also randomly adds new indexes to the chromosome and removes existing indexes. This allows the chromosome to both decrease and increase in size. The crossover operation randomly swaps indexes between two chromosomes. If one chromosome is longer than another, the crossover is done over the first l edge indexes, where l is the length of the shorter chromosome, and the longer chromosome's remaining indexes are randomly spread between the two chromosomes.

When the genetic algorithm terminates, the hash table contains all states seen by the genetic algorithm. Through enumerating the states contained in the hash table, we estimate the expected amount of traffic lost. Specifically, let $X_h \subseteq X$ be the set of all states stored in the hash table. We can obtain a lower bound on $L(N)$ as follows:

$$L(N) \geq \sum_{x \in X_h} p(x) \cdot L(x). \tag{3.8}$$

F. Monte Carlo Method

Monte Carlo simulation has long been the standard approach for estimating network reliability. In general, Monte Carlo simulations suffer from inefficient run times due to sampling non-failure states with high probability. We mitigated this effect by employing a hash table that stores all states encountered so far, similarly to that used by our genetic algorithm. As a result, the speed of the simulation has been significantly improved, which allowed us to execute billions of runs on a fast machine. In addition to increasing the speed, the hash table also adds the advantage of tracking every state which was visited. By the nature of a Monte Carlo simulation, the most likely states will be visited more often, and as such will be contained within the hash table with high probability. This means we can establish lower and upper bounds on the estimated traffic loss by enumerating the visited states.

We can determine a lower bound from the Monte Carlo simulation in the same manner as in Section E. An upper bound can be defined by examining each flow in the original network. If we disallow restoration, then a flow (v, u) will be lost if and only if each of its paths $P^1_{(v,u)}$ and $P^2_{(v,u)}$ has at least one failed edge. The probability that a path has at least one failed edge can be computed as follows:

$$p_f\left(P^i_{(u,v)}\right) = 1 - \prod_{e \in P^i_{(u,v)}} (1 - p(e)) \quad i \in \{1, 2\}. \tag{3.9}$$

An upper bound, $L_u(N)$, on $L(N)$ can be obtained by assuming that the entire traffic demand $M_{(u,v)}$ is lost.

$$L_u(N) = \sum_{(u,v)\in Z} M_{(u,v)} \cdot p_f\left(P^1_{(u,v)}\right) \cdot p_f\left(P^2_{(u,v)}\right). \tag{3.10}$$

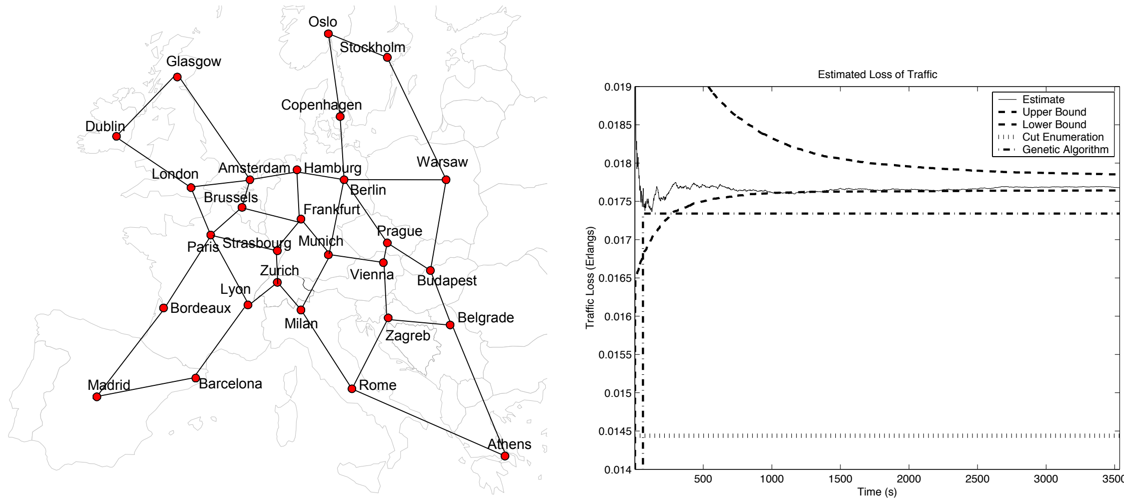
This upper bound can be improved by enumerating states. If a state has a connection where both paths are cut, and it is able to reroute at least part of the traffic, then this is a state where Equation (3.10) is over counting $L(N)$. Let $N(x)$ be the total amount of all traffic that can be rerouted at state x (from all affected connections). Then, we can obtain the following tighter upper bound on $L(N)$:

$$\hat{L}_u(N) = L_u(N) - \sum_{x\in X_h} p(x) \cdot N(x). \tag{3.11}$$

G. Numerical Study

In this section, we present the results of our simulation study. For our simulations, we used the Pan European Network depicted in Figure 12(a) [49]. This network has 28 nodes and 41 edges ranging in length from 218km to 1500km. Using the estimate of 3 cuts per year for every 1000 miles, and a repair time of 14 hours [3], we assumed that the instantaneous probability of failure of edge $e \in E$ is $3/1000 * \frac{14}{24*365} * l(e)$ where $l(e)$ is the length of e. Under this assumption, the probabilities of edge failures in the Pan European Network are in the range of $6.5 * 10^{-4}$ to $4.5 * 10^{-3}$. In this study, we ran our Monte Carlo simulations for 2 billion runs and our Cut Enumeration examined the 1000 most likely cuts. We ran our genetic algorithm for a total of 600 generations and a population size of 200 individuals.

We found that in 2 billion runs of the Monte Carlo simulation, the upper bound was less than 2% above the lower bound as shown in Figure 12(b). The lower bound seemed to converge faster than the upper bound, with the current estimate more



(a) Pan European Network. (b) Monte Carlo Simulation.

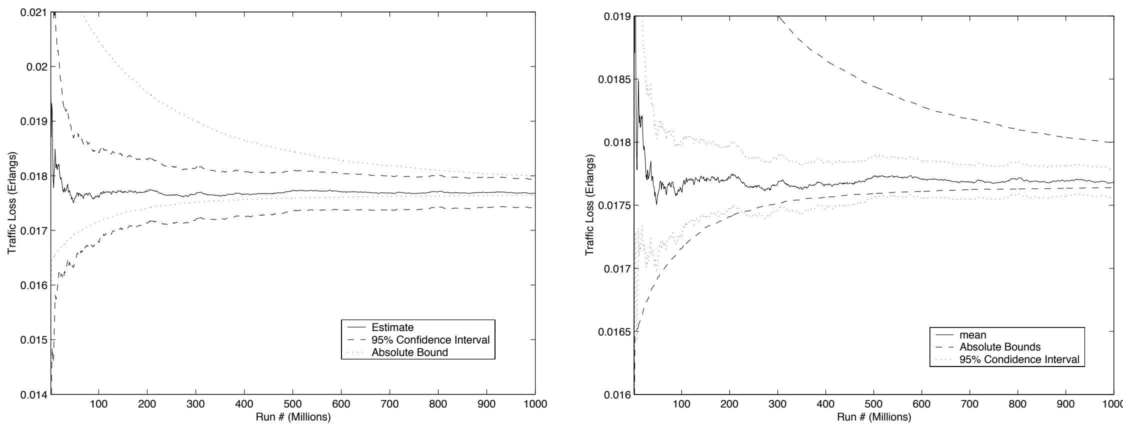
Fig. 12. The Test Case and the Simulation Results.

closely following the lower bound. It is important to note that the Monte Carlo simulation is not constrained to providing an estimate within these bounds, but rather it will provide an accurate estimate with high probability given enough runs.

We can place probabilistic bounds on the mean due to the large number of runs. The first is through the Chebyshev's inequality. By setting the left hand side of the equation to 5%, and solving for ϵ, we can find a 95% confidence bound for the mean. Figure 13(a) shows this confidence bound for the Monte Carlo simulation in comparison to the absolute bounds found through the states in the hash table.

If we assume that the mean is normally distributed, as per the Central Limit Theorem, we can calculate a tighter 95% confidence intervals. Figure 13(b) shows this tighter confidence interval. Here the lower confidence bound is tighter than the absolute bound in earlier runs, but the absolute bound converges more quickly than the confidence bound.

The bound provided by the cut enumeration technique was $1.443639 \cdot 10^{-2}$, which



(a) Monte Carlo Simulation with Chebyshev Confidence Bounds. (b) Monte Carlo Simulation with Normal Confidence Intervals.

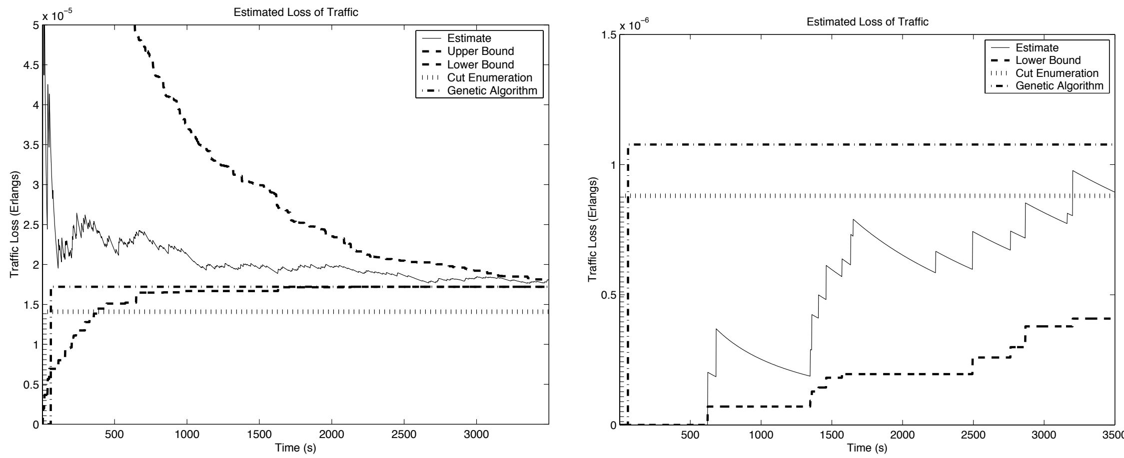
Fig. 13. Simulation Results.

is 18% below the bound found by Monte Carlo. While the cut enumeration technique does not yield a tight bound, it yields it very quickly. The genetic algorithm found a lower bound of $1.734051 \cdot 10^{-2}$, which is 1.7% below the bound found by Monte Carlo.

Next, we reduced the probability of failure of each edge a factor of 32 to see its effect on the simulation. As shown in Figure 14(a), the upper and lower bounds converge much more slowly than the original. This can be explained by the fact that the reduction of probabilities has reduced the probability of cuts resulting in traffic loss therefore they are less likely to be explored. The genetic algorithm yielded a slightly higher bound that Monte Carlo, and took considerably less time.

We further reduced the probability of failure of each edge by 128. As shown in Figure 14(b), the Monte Carlo simulation still has a large range of values to explore. Both the cut enumeration technique and genetic algorithm provide a higher lower bound than the Monte Carlo simulation, with the genetic algorithm being the higher of the two.

On a 3.2GHz Xeon machine, the genetic algorithm took about 60 seconds re-



(a) Monte Carlo Simulation with Probability Reduced by 32.

(b) Monte Carlo Simulation with Probability Reduced by 128.

Fig. 14. Simulation Results With Smaller Probabilities.

gardless of the probability of failure. Running Monte Carlo 2 billion times took approximately 3500 seconds. With the original edge probability, the Monte Carlo achieved a better lower bound than the genetic algorithm after 148 million samples which took 270 seconds. In both cases where we reduced the probability, we allowed Monte Carlo to execute for 2 billion runs, taking over 3500 seconds each time, and it never achieved the lower bound found by the genetic algorithm.

H. Conclusion

In this chapter we presented efficient methods for evaluating the traffic loss in backbone transport networks. We have presented efficient algorithms for evaluation of the expected loss of traffic (ELT). Such a measure is of major importance to service providers since the loss of traffic results in reduced revenues and loss of customer satisfaction. The first method is based on the direct application of Monte Carlo sampling techniques to the problem at hand. The second method is based on combinatorial techniques, specifically enumeration of minimum cuts. The third method employes genetic algorithms together with the niching technique.

Our results show that Monte Carlo algorithms are efficient for networks with a small probability of an edge failure, but their performance falls sharply as the probability of edge failure is reduced. Enumeration of minimum cuts provides a fast lower bound in all cases. The genetic algorithm provides a separate lower bound which is more accurate than the minimum cut enumeration in low probability cases, which are typical in practical settings.

CHAPTER IV

DESIGN OF EFFICIENT ROBUST NETWORK CODES FOR MULTICAST CONNECTIONS

A. Introduction

In recent years, a significant effort has been devoted to improving the resilience of communication networks to failures and increasing their survivability, but edge failures are frequent in communication networks due to the inherent vulnerability of the communication infrastructure [50]. With the dramatic increase in data transmission rates, even a single failure may result in vast data losses and cause major service disruptions for many users. Accordingly, there is a significant interest in network recovery mechanisms that enable a continuous flow of data from the source to the destination with minimal data loss in the event of a failure.

In this chapter, we consider the problem of establishing reliable multicast connections across a communication network with uniform and non-uniform edge capacities. Our goal is to provide *instantaneous recovery* from single edge failures. The instantaneous recovery mechanisms ensure continuous flow of data from the source to the destination node, with no interruption or data loss in the event of a failure. Such mechanisms eliminate the need for packet retransmissions and rerouting. Instantaneous recovery is typically achieved by sending packets over multiple paths in a way that ensures that the destination node can recover the data it needs from the received packets. The three major methods for achieving instantaneous recovery are *dedicated path protection scheme* [50], *diversity coding* [51], and *network coding* [6, 7, 52, 53, 54]. However, only the network coding technique can achieve the optimum in terms of the maximum number of packets that can be sent reliably from the source s to all terminals.

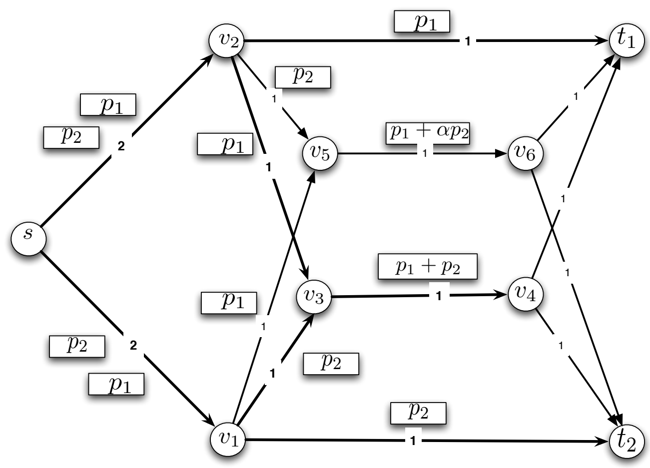


Fig. 15. An Example of an Instantaneous Recovery Scheme With Network Coding.

Consider the multicast network depicted in Figure 15 which needs to deliver two packets per communication round from the source node s to two destination nodes t_1 and t_2. In this example, the source node s needs to deliver two packets, p_1 and p_2 per a communication round, to each terminal with each edge's capacity is as indicated. In this network, edges (s, v_1) and (s, v_2) can send two packets per communication round, while all other edges can send only one packet per communication round. Here, α is an element of a finite field different from one. The network code requires a field of size at least three. The figure shows an encoding scheme that delivers two packets p_1 and p_2 to terminals t_1 and t_2 over a single round such that both destination nodes can decode the packets sent by the source node in any single edge failure scenario. Note that without the encoding operation at the intermediate nodes v_3 and v_5, it would not be not possible to send two packets with instantaneous recovery.

Related Work

The network coding technique has been introduced in the seminal paper of Ahlswede et al. [6]. Initial work on network coding has focused on multicast connections. It was shown in [6] that the maximum rate of a multicast network is equal to the minimum total capacity of a cut that separates the source from a terminal. This maximum rate can be achieved by using linear network codes [55]. Koetter and Médard [7] developed an algebraic framework for linear network codes. Ho et al. [56] showed that the maximum rate can be achieved by using random linear network codes. Jaggi et al. [57] proposed a deterministic polynomial-time algorithm for finding feasible network codes in multicast networks. Network coding algorithms resilient to malicious interference have been studied in [58], [59], and [60]. Comprehensive surveys on the network coding techniques are available in the recent books [61, 62], and [63].

The idea of using network coding for instantaneous recovery from edge failures was first described by Koetter and Medard [7]. They showed that if the network has a sufficient capacity to recover from each failure scenario (e.g., by rerouting) then instantaneous recovery from each failure scenario can be achieved by employing linear network codes. Ho et al. [52] presented an information-theoretic framework for network management in the presence of edge failures. Using network coding for reliable communication was also discussed in [53] and [54]. References [64] and [65] describe practical implementations of network coding and demonstrate its benefits for improving reliability and robustness in communication networks.

1. Our Contribution

In this chapter we propose efficient algorithms for construction of robust network codes over small finite fields. We consider two major cases. In the first case, we assume that all edges of the network have uniform capacity, while the second case allows the capacity of network edges can vary. For the first case we present an efficient network coding algorithm that identifies a robust network code over a small field. The algorithm takes advantage of special properties of Maximum Rank Distance (MRD) codes [66]. For the second case, we focus on settings in which the source node needs to deliver two packets per time unit to all terminals. We show that in this case, a special topological properties of robust coding networks can be exploited for constructing a network code over a small finite field.

A robust network code for multicast networks can be established through the standard network coding algorithm presented in [57]. However, this algorithm is designed to handle arbitrary failure patters and, as a result, requires a field size of $O(|E|)$ in the case of single edge failures, where E is the set of network edges. In contrast, our scheme requires a small field size ($O(k)$, where k is the number of terminals), which does not depend on the size of the underlying communication network. The size of the finite field is a very important factor in practical implementation schemes [64] as it determines the amount of communication and computational overhead. In addition, the computational complexity of our algorithm is smaller than that of the existing solutions.

B. Model

1. Multicast Network

We consider a multicast network, $\mathbb{N}$, that uses a directed acyclic graph, $G(V, E)$, to send data from the source, s, to a set, T, of k destination nodes, $\{t_1, \ldots, t_k\} \subset V$. The data is delivered in packets, each an element of a finite field, $\mathbb{F}_q = GF(q)$. We also assume that the data exchange is performed in rounds, such that each edge,

$e \in E$, can transmit $c(e)$ packets per communication round. We assume that $c(e)$ is an integer number and refer to it as the *capacity* of edge e. At each communication round, the source node needs to transmit h packets, $\mathbb{R} = (p_1, p_2, \dots, p_h)^T$, from the source node, $s \in V$, to each destination node, $t \in T$. We refer to h as the *rate* of the multicast connection. It was shown in [6] and [55] that the maximum rate of the network, i.e., the maximum number of packets that can be sent from the source, s, to a set, T, of terminals per time unit, is equal to the minimum capacity of a cut that separates the source, s, from a terminal, $t \in T$. Accordingly, we say that a multicast network, $\mathbb{N}$, is *feasible* if any cut that separates s and a terminal, $t \in T$, has at least h edges. We say that a coding network, $\mathbb{N}$, is minimal if any network formed from $\mathbb{N}$ by removing an edge or decreasing the capacity of an edge is no longer feasible. It is easy to verify that the capacity of each edge in a minimal network is bounded by h.

2. Coding Networks

For clarity of presentation, we define an auxiliary graph, $\hat{G}(V, A)$, formed by the network graph $G(V, E)$ by substituting each edge, $e \in E$, by $c(e)$ parallel *arcs* that have the same tail and head nodes as e; each arc can transmit one packet per communication round. We denote $A(e) \subseteq A$ as the set of arcs that correspond to edge e. In what follows we only refer to packets sent in the current communication round. The packets sent in the subsequent rounds are handled in a similar manner.

A network code is defined by associating each arc, $a(v, u) \in A$, in the network with a *local encoding* function f_a. The local encoding function specifies the packet transmitted by arc a as a function of the packets available at or received by the tail node of a in the current communication round. More specifically, for each outgoing arc, $a(s, u) \in A$, of the source node s, f_a is a function of the original h packets, $\mathbb{P}$, i.e., $f_a : \mathbb{F}^h \to \mathbb{F}$. For any other arc, $a(v, u) \in A$, $v \neq s$, f_a is a function of the packets received by node v in the current round, i.e., $f_a : \mathbb{F}^l \to \mathbb{F}$, where l is the number of incoming arcs of v in $\hat{G}$. A network code, $\mathbb{C}$, is a set of encoding functions associated with the arcs in A, i.e, $\mathbb{C} = \{f_a \mid a \in A\}$. In a *linear network code* all packets are elements of a finite field and all local encoding functions are linear functions over that field.

3. Robust Coding Networks

As mentioned in the introduction, we assume that only one of the edges in the network can fail at any time. Since a failed edge, e, cannot transmit packets, we assume that the encoding function f_a of each arc, $a \in A(e)$ is identically equal to zero, i.e., $f_a \equiv \mathbf{0}$. To guarantee instantaneous recovery, it is sufficient to ensure that for each edge failure, there exists a set of h linearly independent packets received by t.

We distinguish between two types of robust networks codes. In *strongly* robust network codes, the local encoding coefficients of all arcs in A remain the same, except for the arcs $A(e)$ that correspond to the failed edge, e, which are assigned zero encoding coefficients. In *weakly* robust network codes, the arcs that are located downstream of the failed edge, e, are allowed to change their encoding coefficients, while all the encoding coefficients that correspond to other edges must remain the same.

Definition 4 (Strongly Robust Network Code) *A network code, $\mathbb{C}$, is said to be* strongly robust *if for each $e \in E$ it holds that the network code $\mathbb{C}'$ formed from $\mathbb{C}$ by assigning zero encoding coefficients to arcs in $A(e)$ is feasible.*

We proceed to discuss weakly robust network codes. When an edge, $e(v, u) \in E$, fails, the set of nodes in the network can be divided into two subsets, V'_e and V''_e. The set V''_e includes all descendants of the node, u, whose head is e, while the set $V'_e =$

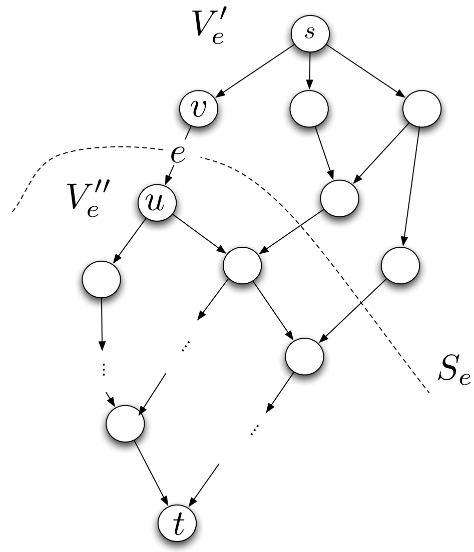


Fig. 16. A Coding Network With a Failed Edge e.

$V \setminus V''_e$ includes all ancestors of the node v, whose tail is e, as well as all other nodes not included in V''_e. Figure 16 shows an example of a cut (V'_e, V''_e) in a coding network. We assume that node u can detect the failure of edge e and notify its immediate descendants, which in turn can change their encoding coefficients so that each affected terminal will be able to decode the original packets. Changing encoding coefficients for the nodes in V''_e can be done with minimum penalty because the information about the failure can be attached to the packets that carry the information. In contrast, in order to modify the network code for arcs that originate from nodes in V'_e we need to send special control messages, which will incur additional delay.

Definition 5 (Weakly Robust Network Code) *A network code, $\mathbb{C}$, is* weakly robust *to single edge failures if for each edge, $e \in E$, there exists a feasible code $\mathbb{C}'$ that satisfies the following three conditions:*

1. *The encoding coefficients of all arcs that originate from nodes in V'_e have the*

the same encoding coefficients in $\mathbb{C}'$ as in $\mathbb{C}$ (except arcs in $A(E)$)

2. *The global encoding coefficients for all arcs in $A(e)$ have zero encoding coefficients*

3. *Each terminal $t \in T$ can decode the original packets.*

4. Necessary Condition

A necessary condition for existence of robust network codes (both weakly and strongly robust) is that for each $e \in E$ a network, G^e, formed from G by removing e, must admit an (s,t)-flow of value h. This condition is equivalent to

$$\min_{C} \left[\sum_{e \in E(C)} c(e) - \max_{e \in E(C)} c(e) \right] \geq h, \tag{4.1}$$

where the minimum is taken over all (s,t)-cuts, $C(V_1, V_2)$, that separate s and t in G, and $E(C)$ is the set of edges that belong to C, i.e., the set of edges that connect a node in V_1 to a node in V_2. In [7] it was shown that this condition is also sufficient for providing instantaneous recovery from edge failures. Moreover, it was shown that the instantaneous recovery can be achieved by using *linear* network codes. Therefore, we refer to a graph $G(V,E)$ that satisfies this condition as a *feasible* graph or network.

C. Strongly Robust Codes for Networks With Uniform Capacities

In this section we assume that all edges of the network have uniform capacity c, i.e., each edge can send exactly c packets per time unit. We present an efficient algorithm that can construct a robust network code over a finite field of size $O(k)$. We observe that without loss of generality, we can assume that the capacity of each edge is one unit. A feasible network code for unit capacity edges can be extended into the case in which the capacity of each edges is equal to c by combining c communication rounds into a single round. Accordingly, for the rest of this section, we assume that all edges have unit capacity.

In [57], it was shown that communications at rate h with instantaneous recovery from single edge failures is possible if and only if for each edge, $e \in E$, it holds that the network $G'(V', E')$ formed by the removal of e from $G(V, E)$, contains at least h edge-disjoint paths from the source, s, to each terminal node, $t \in T$. This implies that a necessary and sufficient condition for the feasibility of the network, $\mathbb{N}$, is the existence of $h+1$ edge-disjoint paths between s and each $t \in T$.

Our approach can be summarized as follows: First, we generate a special parity check packet, referred to as p_{h+1}. This packet is a linear combination of the original packets and is constructed as described in Section 1. Then, we use a standard network coding algorithm due to Jaggi et al. [57] for sending $\hat{\mathbb{R}} = \{p_1, p_2, \ldots, p_h, p_{h+1}\}$ packets from s to T.

The standard algorithm will treat the packets in $\hat{\mathbb{R}}$ as generated by independent random processes. The algorithm ensures that, in the normal network conditions, each destination node receives $h+1$ independent linear combination of the packets in $\hat{\mathbb{R}}$. The following lemma shows that after a single edge failure, each destination node receives at least h linearly independent combinations of packets in $\hat{\mathbb{R}}$.

Lemma 6 *Upon an edge failure, each terminal $t \in T$ receives at least h linear combinations of packets in $\hat{\mathbb{R}}$.*

Proof: Since we assume that all edges are of unit capacity, so each edge in the network can be represented by a single arc. For each arc, $a \in A$, we define the *global encoding vector*

$$\Gamma_e = [\gamma_1^e \ \ldots \ \gamma_{h+1}^e]^T \in \mathbf{F}_q^{h+1},$$

that captures the relation between the packet, p_a, transmitted on arc a and the original packets in $\hat{\mathbb{R}}$:

$$p_e = \sum_{i=1}^{h+1} p_i \cdot \gamma_i^e. \tag{4.2}$$

Let t be a terminal in T. We define the *transfer matrix*, M_t, that captures the relation between the original packets, $\mathbb{R}$, and the packets received by the terminal node, $t \in T$, over its incoming edges. The matrix M_t is defined as follows:

$$M_t = \begin{bmatrix} \Gamma_{a_t^1} & \Gamma_{a_t^2} & \dots & \Gamma_{a_t^{h+1}} \end{bmatrix}, \tag{4.3}$$

where $E_t = \{a_t^1, \dots, a_t^{h+1}\}$ is the set of incoming arcs of terminal t.

Let a' be a failed arc. A failure of a' might result in a change of the the transfer matrix M_t. We note that the new network matrix $\hat{M}_t$ can be written as:

$$\hat{M}_t = M_t - \Gamma_{a'} T_{a'},$$

where $T_{a'}$ is an $1 \times h$ matrix that depends on the location of arc a' in the network. Note that $\Gamma_{a'} T_{a'}$ is an $(h+1) \times (h+1)$ matrix of rank no more than one. The subadditivity property[1] of rank implies that the rank of $\hat{M}_t$ is at least h. ■

1. Creating Parity Check Packet

Lemma 6 implies that in the event of any single edge failure, each terminal node receives at least h independent linear combinations of the packets in $\{p_1, p_2, \dots, p_h, p_{h+1}\}$. Since packet p_{h+1} is a linear combination of h original packets $\mathbb{R} = \{p_1, p_2, \dots, p_h\}$, each destination nodes receives, in fact, h linear combinations of $\mathbb{R}$. Accordingly, our goal is to construct packet p_{h+1} in such a way that each destination node receives h independent linear combinations of $\mathbb{R}$. This will allow each destination node to

[1]The subadditivity property of rank implies that $Rank(X + Y) \leq Rank(X) + Rank(Y)$.

decode the original packets.

For clarity of presentation we first focus on the case of $h = 2$. In this case we have two original packets, p_1 and p_2, and one parity check packet $p_3 = \gamma_1 \cdot p_1 + \gamma_2 \cdot p_2$. Suppose that a terminal $t \in T$ receives two linearly independent combinations of p_1, p_2, and p_3.

There are three different forms of packets a sink may receive, differing by their coefficients for packet p_3. In the first case, a sink receives two packets which both have zero coefficients for p_3. In this case, decoding of p_1 and p_2 is trivial. If one packet has a non-zero coefficient for p_3, and the second has a zero coefficient for p_3, we can express the packets as in Equation 4.4 below by dividing the first packet by its coefficient for p_3:

$$\begin{bmatrix} \beta_1 & \beta_2 & 1 \\ \beta_3 & \beta_4 & 0 \end{bmatrix} \times \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}, \tag{4.4}$$

where $\beta_1, \ldots, \beta_4$ are coefficients that belong to $\mathbb{F}_q$. By substituting $p_3 = \gamma_1 \cdot p_1 + \gamma_2 \cdot p_2$ we get:

$$\begin{bmatrix} (\beta_1 + \gamma_1) & (\beta_2 + \gamma_2) \\ \beta_3 & \beta_4 \end{bmatrix} \times \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}. \tag{4.5}$$

We wish to find values for γ_1 and γ_2 such that matrix

$$R = \begin{bmatrix} (\beta_1 + \gamma_1) & (\beta_2 + \gamma_2) \\ \beta_3 & \beta_4 \end{bmatrix}$$

is of full rank. In particular, our goal is to select the values of γ_1 and γ_2 in such a way that the determinant

$$\det(R) = \beta_1\beta_4 - \beta_2\beta_3 + \gamma_1\beta_4 - \gamma_2\beta_3$$

of R is not equal to zero. Since no constraints are made on the encoding coefficients used in the multicast network code, $\det(R)$ must not be zero for all possible choices of $\beta_1, \ldots, \beta_4$ where the terminal receives two linearly independent vectors. It is easy to verify that this cannot be done if γ_1 and γ_2 belong to the same field, $\mathbb{F}_q$, as $\beta_1, \ldots, \beta_4$.

Accordingly, in our approach we select γ_1 and γ_2 in an extension field of $\mathbb{F}_q$ and construct the parity check packet p_3 using an MRD code [66].

In an MRD code, one uses a vector of elements in $GF(q)$ to create an element in an extension field. A vector of N elements from $GF(q)$ can be treated as an element in $GF(q^N)$ where $GF(q^N)$ is an extension field of $GF(q)$. A (n, m) MRD code over $GF(q^N)$ takes m information symbols and generates n encoded symbols. It is capable of correcting $n - m$ rank erasures, or otherwise stated, can recover m information packets from any m linearly independent combinations of the encoded packets [66]. Since we are interested in recovery of the packets of p_1 and p_2 from any two linearly independent packets originating from p_1, p_2, and p_3, a $(3, 2)$ MRD code is sufficient. Such a code can be constructed using the following parity check matrix [67]:

$$H = \begin{bmatrix} \alpha & \alpha^2 & 1 \end{bmatrix},$$

where α is the primitive element of $GF(q^3)$. Using H, we can set $p_3 = -\alpha p_1 - \alpha^2 p_2$. Note that this way we are forced to work in the extension field, $GF(q^3)$, of $\mathbb{F}_q = GF(q)$. This implies that the original packets p_1, p_2, p_3 must belong to $GF(q^3)$ as well. This can be achieved by combining three communication rounds into a single round and treating a vector of size three in $GF(q)$ as a single element of $GF(q^3)$.

If we re-examine R, we see that elements from $GF(q)$ are mixed with elements from $GF(q^3)$. Since $GF(q^3)$ is constructed as an extension field of $GF(q)$, $\beta_1, \ldots, \beta_4$ can be treated as elements from a subset of $GF(q^3)$ which behave as elements from

$GF(q)$. This implies that $\beta_1\beta_4 - \beta_2\beta_3$ belongs to $GF(q)$, and $\alpha\beta_4 - \alpha^2\beta_3$ might also be in $GF(q)$ if $\alpha - \alpha^2 \in GF(q)$. However, this is not the case since $\begin{bmatrix} 1 & \alpha & \alpha^2 \end{bmatrix}$ is a valid MRD code[66]. Therefore,

$$\det(R) = \beta_1\beta_4 - \beta_2\beta_3 + \alpha\beta_4 - \alpha^2\beta_3$$

is equal to zero only if $\beta_3 = \beta_4 = 0$, which contradicts our assumption that the destination node receives two linearly independent vectors.

Lastly, we consider the case where both received vectors contain non-zero coefficients of p_3, as in Equation 4.6. Through Gaussian elimination, the two vectors can be subtracted to construct vectors as in Equation 4.4, and by using the argument described above we can show that packets p_1 and p_2 can be recovered as well.

$$\begin{bmatrix} \beta_1 & \beta_2 & 1 \\ \beta_3 & \beta_4 & 1 \end{bmatrix} \times \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}. \tag{4.6}$$

2. General Case

We turn to consider a more general case of $h > 2$. Since we need to only recover from a single failure, we need to find a $(h+1, h)$ MRD code. For this case, we can use the parity check matrix [66]:

$$H = \begin{bmatrix} \alpha & \alpha^2 & \dots & \alpha^h & 1 \end{bmatrix} \tag{4.7}$$

over the field $GF(q^{h+1})$, with

$$p_{h+1} = -\sum_{i=1}^{h} \alpha^i p_i. \tag{4.8}$$

We summarize our results by the following theorem:

Theorem 7 *The proposed scheme achieves an instantaneous recovery from any single edge failure.*

Proof: Follows directly from the properties of MRD codes [66]. ■

D. Network With Non-Uniform Capacities

In this section, we assume that different network edges may have different capacities. We focus on a special case in which only two packets need to be delivered from the source to all terminals at each communication round. The design of robust network codes for $h = 2$ in the context of unicast connections has been studied in [68]. In this work, we build on the results of [68] for constructing a robust network code for multicast connections.

Let $G(V, E)$ be a minimum robust coding network, i.e, a feasible robust network such that the removal of an edge or a reduction in the capacity of an edge results in a violation of its feasibility. Note that the capacity of any edge, $e \in E$, is at most two. For each terminal, $t \in T$, let $G_t(V_t, E_t)$ to be a subgraph of $G(V, E)$ that contains a minimum coding network with respect to terminal t. That is, $G_t(V_t, E_t)$ only contains edges of $G(V, E)$ that are necessary to guarantee the conditions defined by Equation 4.1 for terminal t. Furthermore, any reduction of the capacity of edges in $G_t(V_t, E_t)$ will result in a violation of this condition for at least one of the (s, t) cuts.

For each $t \in T$, we denote A_t as the set of arcs that correspond to edges in E_t. In [68] it was shown that it is possible to construct a robust unicast network code over $GF(2)$. In this code, each arc transmits either one of the original packets, p_1 and p_2, or their sum, $(p_1 + p_2)$. For each $t \in T$ we denote such network code as $\mathbb{C}_t$. We use $\mathbb{C}_t$ to divide A_t into three disjoint subsets A_t^1, A_t^2, and A_t^3, where all the arcs in A_t^i

carry the same packet.

To extend this to multicast, we construct a set of $f = 3^k$ linearly independent vectors $\Phi = \{\Phi^1, \Phi^2, \dots, \Phi^f\}$ of the original two packets over $GF(q)$.

We divide the arcs in A into f different subsets $A^1, ..., A^f$ such that for each subset, A^i all arcs in A^i belong to the same subsets, A_t^j, for all t when the the arc belongs to A_t. That is, for any two arcs, a_1 and a_2, in A^i and for each terminal t such that $a_1, a_2 \in A_t$, both a_1 and a_2 belong the same subset, say A_t^j, of A_t. Then each subset A^i is associated with a linearly independent vector, Φ^i, from $GF(q)$. We construct a code for which it holds that the global encoding coefficient of each arc in A^i is equal to Φ^i. It is easy to verify that such a network code is feasible, i.e., it is possible to select the set of local encoding coefficients that satisfy this property. The proof is based on the fact that for $h = 2$, any two linearly independent packets are sufficient for constructing any linear combination in Φ. Also, it is easy to verify that the code has a weakly robust property.

We construct the multicast code by creating a vector of t elements, each element taking on any of 3 values, for each arc $a \in A$. We map elements in A_t^1 to have a value of 1 in index t. The same is true for A_t^2 and A_t^3 being mapped to 2 and 3 respectively. If $a \ni A_t$, its value at index t is not constrained. Each unique vector corresponds to a subset, A^i, which is assigned to codeword Φ^i. Codewords are assigned in topological order, assigning any non-constrained values any feasible values.

Lemma 8 *The multicast code construction is a feasible multicast code*

Proof: First we consider coding nodes within the network. If incident arcs to a coding node contain two linearly independent packets, then any packet may be constructed for outgoing arcs. If all incoming arcs contain the same packet, then we show that the outgoing packets are the same under code construction. We need only consider constrained indexes on an outgoing arcs, i.e. indexes where $\mathbb{C}_t \neq 0$ since non-constrained entries may take on any value. Since the incoming arcs contain the same packet, the arcs must contain a single codeword or the 0 packet in $\mathbb{C}_t$. In $\mathbb{C}_t$, the outgoing packets may only contain the same codeword, or the 0 packet. In both cases, the same incoming multicast packet is valid. Lastly, the case where an incoming arc contains the 0 packet is the same as a network where the incoming arc does not exist. Next we consider packets received by each sink. For each sink $t \in T$, $\mathbb{C}_t$ guarantees at least arcs a_1 and a_2 whose head is t, carry linearly independent packets in the unicast case. These two arcs belong to different subsets, A_t^i and A_t^j, and belong to different sets, A^k and A^l, by construction. Since these two sets corresponds to different codewords, Φ^k and Φ^l, they are linearly independent. ■

Lemma 9 *The multicast code construction is weakly robust*

Proof: First we note in any case where two arcs contain linearly independent packets in $\mathbb{C}_t$, they also contain linearly independent packets in the multicast code. Second, we note that each $\mathbb{C}_t$ results in two linearly independent packets received by t in the presence of a single edge failure. We define $\mathbb{C}'_t$ to define the global coding coefficients in the unicast code after an edge failure. Note, only descendants of the failed edge contain different coefficients in $\mathbb{C}_t$ and $\mathbb{C}'_t$. In our multicast code, we change the packet on the failed edge to the 0 packet and substitute $\mathbb{C}'_t$ for $\mathbb{C}_t$. Since descendants of the failed edge can be informed of the failure with received packets, they can make such a substitution. Lastly, since $\mathbb{C}'_t$ is feasible, the multicast code is still feasible in the presence of a single edge failure, making it weakly robust. ■

E. Conclusion

In this chapter, we considered the problem of establishing reliable multicast connections across a communication network. For the case of uniform edge capacities, we presented an efficient network coding algorithm based on the MRD codes that requires a small finite field $O(k)$. For the case of non-uniform capacities, we focused on a special case of $h = 2$ and showed that it is also possible to construct a robust code over a small field. Future research includes the construction of network codes with non-uniform capacities and transmission of more than two packets per communication round.

CHAPTER V

A RANDOMIZED ALGORITHM AND PERFORMANCE BOUNDS FOR CODED COOPERATIVE DATA EXCHANGE

A. Introduction

The ever-growing demand of mobile wireless clients for large file downloads and video applications is straining cellular networks in terms of bandwidth provision and network cost. Inspired by the Internet paradigm where peer-to-peer (P2P) content delivery systems are more efficient than a server-client based model, one solution to address these issues is to allow the mobile clients to cooperate and exchange data directly among each other.

In this chapter, we consider the problem of the information exchange between a group of wireless clients. Each client initially holds a subset of packets and needs to obtain all the packets held by other clients. Each client can broadcast the packets in its possession (or a combination thereof) via a noiseless broadcast channel of capacity one packet per channel use. Assuming that clients can cooperate with each other and are fully aware of the packets available to other clients, the aim is to minimize the total number of transmissions needed to satisfy the demands of all clients.

For example, Fig. 17 shows three wireless clients that are interested in obtaining three packets of m bits each, x_1, x_2 and $x_3 \in GF(2^m)$. The first, second and third clients have already obtained packets $\{x_2, x_3\}$, $\{x_1, x_3\}$ and $\{x_1, x_2\}$, respectively, i.e., each of these clients misses one packet. A simple cooperation scheme would consist of three uncoded transmissions. However, this is not an optimal solution since the clients can send coded packets and help multiple clients with a single transmission. The number of transmissions for this example can be decreased to two by letting the

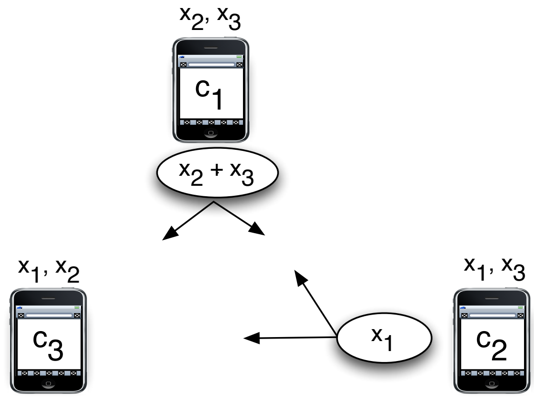


Fig. 17. Coded Data Exchange Among Three Clients.

first client send $x_2 + x_3$ and the second client send x_1.

The problem we consider may appear in many practical settings. For example, consider a wireless network in which some clients are interested in the same data (such as a popular video clip or an urgent alert message). Initially, the entire data is available at a base station and is broadcast to the interested clients. The communication link between the base station and the mobile clients can be not only expensive, but also unreliable or sometimes even non-existent, which causes some clients to receive only a portion of the data. In particular, partial reception can be caused by *channel fading or shadowing*, *connection loss*, *network saturation*, or *asynchronous client behavior* such as in P2P systems. Despite partial reception, whenever the whole data is collectively known by the interested clients, they can help each other to acquire the whole data using short-range client-to-client communication links or cooperative relaying which can be more affordable or reliable.

In this chapter we investigate theoretical aspects of such client cooperation and are interested in finding efficient data exchange strategies which require minimum total number of transmissions. This problem was introduced by Rouayheb et al. in [8] where lower and upper bounds on the minimum number of transmissions were presented, in addition to a data exchange algorithm. Building on [8], we propose an optimal data exchange algorithm based on random linear coding over a large field and then show how coding can be done over a smaller field, once the number of transmissions from each client is known.

A closely-related problem is that of Index Coding [69, 70, 71] in which different clients cannot communicate with each other, but can receive transmissions from a server possessing all the data. Gossip algorithms [72] and physical layer cooperation [73] are also related concepts which are extensively studied in the literature.

B. System Model

Consider a set of n packets, $X = \{x_1, \ldots, x_n\}$, to be delivered to k clients belonging to the set $C = \{c_1, \ldots, c_k\}$. The packets are elements of a finite alphabet which will be assumed to be a finite field, $\mathbb{F}$, throughout this chapter. At the beginning, each client knows a subset of packets denoted by $X_i \subseteq X$, while the clients collectively know all packets in X, i.e., $\cup_{c_i \in C} X_i = X$. We denote $\overline{X}_i = X \setminus X_i$ as the set of packets required by client c_i. We assume that each client knows the index of packets that are available to other clients.[1]

The clients exchange packets over a lossless broadcast channel with the purpose of making all packets in X available to all clients. The data is transferred in communication rounds, such that at round i one of the clients, say c_j, broadcasts a packet, $p_i \in \mathbb{F}$, to the rest of the clients in C. Packet p_i may be one of the packets in X_j, or a combination of packets in X_j and the packets $\{p_1, \ldots, p_{i-1}\}$ previously transmitted

[1]This can be achieved by exchanging packet indices at the beginning of data exchange. The indices can also be piggybacked on the data packets to reduce overhead.

over the channel. Our goal is to devise a scheme that enables each client $c_i \in C$ to obtain all packets in $\overline{X}_i$ while minimizing the total number of transmissions. Our schemes use linear coding over the field $\mathbb{F}$. As discussed in Section D below, restricting ourselves to linear coding operations does not result in loss of optimality.

With linear coding, any packet, p_i, transmitted by the algorithm is a linear combination of the original packets in X, i.e.,

$$p_i = \sum_{x_j \in X} \gamma_i^j x_j,$$

where $\gamma_i^j \in \mathbb{F}$ are *encoding coefficients* of p_i.We refer to the vector $\gamma_i = [\gamma_i^1, \gamma_i^2, \ldots, \gamma_i^n]$ as the *encoding vector* of p_i. The i-th *unit* encoding vector that corresponds to the original packet x_i is denoted by $u_i = [u_i^1, u_i^2, \ldots, u_i^n]$, where $u_i^i = 1$ and $u_i^j = 0$ for $i \neq j$. We also denote U_i as the set of unit vectors that corresponds to packets in X_i. Note that we are restricting the transmission by client c_j to being a linear combination of packets initially known to client c_j and ignoring packets already broadcasted on the channel. It can be shown that this will have no effect on the optimality of the algorithm.

Let $n_i = |X_i|$ be the number of packets initially known to client c_i. The number of unknown packets to client c_i is therefore, $\bar{n}_i = |\overline{X_i}| = n - n_i$. We denote $n_{\min} = \min_{1 \leq i \leq k} n_i$ as the minimum number of packets known to a client. The corresponding client or clients where $n_i = n_{\min}$ form a subset $C_{\min}$ of C.

A client c_i is said to have a *unique* packet x_j if $x_j \in X_i$ and $x_j \notin X_\ell$ for all $\ell \neq i$. A unique packet can be broadcast by the client holding it in an uncoded fashion at any stage without any penalty in terms of optimality. Without loss of generality, we can assume that there are no unique packets in the system. Additionally, without loss of generality, we assume that all k clients initially have distinct packet sets.

Algorithm Information Exchange (k, n, Γ, C)
For $i \leftarrow 1$ to k **do**
    $Y_i = \langle\{\Gamma_x \mid x \in X_i\}\rangle$
While there is a client i with $\dim Y_i < n$ **do**
    While $\exists c_i, c_j \in C$ $i \neq j$, such that $Y_i = Y_j$ **do**
        $C = C \setminus \{c_i\}$
    Find a client c_i with a subspace Y_i of maximum dimension
    (If there are multiple such clients choose an arbitrary one of them)
    Select a vector $b \in Y_i$ such that $b \notin Y_j$ for each $j \neq i$
    Let client c_i broadcast packet $b \cdot (x_1, \ldots, x_n)^T$
    For $\ell = 1 \leftarrow 1$ to k **do**
        $Y_i \leftarrow Y_i + \langle\{b\}\rangle$

Fig. 18. Algorithm Information Exchange

We note that the results of this chapter can be applied, with minor modifications, to settings where initial data available to clients include linear combinations of the packets in X. However, these settings are beyond the scope of this chapter.

C. Deterministic Algorithm for Three Clients

1. Optimality Proof

Algorithm Information Exchange presented by Rouayheb et al. in [8] shown in Figure 18, is optimal for two and three clients with slight modification. In the case of two clients, each client must transmit the other client's compliment set. It is easy to show that the proposed algorithm accomplishes this task. When run on three clients, the algorithm must be modified to transmit all unique packets first. Since the original algorithm assumes unique packets, their presence can interfere with its optimality if they are not initially handled. See the example in Subsection 3 below with four clients as a demonstration. For clarity, we assume that Algorithm Information Exchange contains this modification.

We first allow the transmission of all unique packets. A unique packet, in terms of vector spaces, we defined as a packet which is orthogonal to all other packets held by other clients. Then, the algorithm proceeds in rounds or iterations, starting with iteration 0. The three clients are assigned labels a, b, and c.[2] Then, we define the vector space spanned by clients a, b, and c at iteration i as A_i, B_i, and C_i, respectively. For clarity, we remove any vector space which is shared by all clients, $A_0 \cap B_0 \cap C_0$. The remaining subspace is defined as $N = A_i + B_i + C_i$, which is the space spanned by the combination of all vectors in A_i, B_i, and C_i.

We define four sets of subspaces:

$$W_i = A_i \cap B_i \cap C_i, \tag{5.1}$$

$$\overline{A}_i = B_i \cap C_i \cap A_i^{\perp}, \tag{5.2}$$

$$\overline{B}_i = A_i \cap C_i \cap B_i^{\perp}, \tag{5.3}$$

$$\overline{C}_i = A_i \cap B_i \cap C_i^{\perp}, \tag{5.4}$$

where $Y^{\perp}$ denotes the subspace orthogonal to subspace Y. Intuitively, $\overline{A}_i$, $\overline{B}_i$, and $\overline{C}_i$ denote subspaces unknown to clients A_i, B_i, and C_i, respectively. Thus, $\dim(\overline{A}_i) + \dim(A_i) = n$ and the same holds for B_i and C_i.

Initially $\overline{A}_0 = B_0 \cap C_0$, $\overline{B}_0 = A_0 \cap C_0$, $\overline{C}_0 = A_0 \cap B_0$ and $W_0 = 0$. By definition $\overline{A}_i \cap \overline{B}_i = 0$, $\overline{A}_i \cap \overline{C}_i = 0$, and $\overline{B}_i \cap \overline{C}_i = 0$ Also, since no subspace is known to only a single client, $A_i = \overline{B}_i + \overline{C}_i + W_i$, $B_i = \overline{A}_i + \overline{C}_i + W_i$, and $C_i = \overline{A}_i + \overline{B}_i + W_i$. Since $\overline{A}_i$, $\overline{B}_i$, $\overline{C}_i$, and W_i form non-overlapping subspaces, it holds that $\dim(N) = \dim(\overline{A}_i) + \dim(\overline{B}_i) + \dim(\overline{C}_i) + \dim(W_i)$.

[2]As explained below, clients are assigned labels at the beginning of each iteration. A client might have different labels in different iterations.

Without loss of generality, for each $i = 0, 1, 2, \ldots,$ we assume that:

$$\dim(\overline{A}_i) \geq \dim(\overline{B}_i) \geq \dim(\overline{C}_i). \tag{5.5}$$

Indeed, we can always rename clients in the beginning of each iteration such that Equation 5.5 is satisfied.

In analysis of this problem, instances of the problem can be separated into two sets based on whether an inequality, which bears similarity to the triangle inequality, is satisfied:

$$\dim(\overline{A}_0) \leq \dim(\overline{B}_0) + \dim(\overline{C}_0). \tag{5.6}$$

First we show that the algorithm is optimal when the triangle inequality is not met.

Lemma 10 *Algorithm* INFORMATION EXCHANGE *is optimal for three clients when* $\dim(\overline{A}_0) > \dim(\overline{B}_0) + \dim(\overline{C}_0)$.

Proof: When $\dim(\overline{A}_0) > \dim(\overline{B}_0) + \dim(\overline{C}_0)$, a trivial lower bound on the number of packet transmissions is $\dim(\overline{A}_0)$ as $\overline{A}_0$ contains the subspace initially unknown to client a.

We prove that for at each iteration i of the algorithm it holds that

$$\dim(\overline{A}_i) > \dim(\overline{B}_i) + \dim(\overline{C}_i).$$

The condition of the lemma implies that this is true at for $i = 0$.

Suppose that this condition is true for iteration i, we prove that it is true for iteration $i + 1$. Algorithm INFORMATION EXCHANGE requires at each iteration the client with the highest packet count transmits a packet which is new to each other client. Then, client c will transmit a packet p_i that must satisfy the following three conditions:

$$
\begin{aligned}
p_i &\in \overline{A}_i + \overline{B}_i, \\
p_i &\notin \overline{A}_i, \\
p_i &\notin \overline{B}_i
\end{aligned} \tag{5.7}
$$

That is, p_i must be within the subspace known to c, but also outside the subspace known to both clients a and b individually. Since $\overline{A}_i$ and $\overline{B}_i$ define a subspace unknown to clients a and b respectively, and packet p_i has been transmitted to clients a and b, it holds that:

$$\overline{A}_{i+1} = \overline{A}_i \cap p_i^{\perp}; \tag{5.8}$$

$$\overline{B}_{i+1} = \overline{B}_i \cap p_i^{\perp}; \tag{5.9}$$

This implies that $\dim(\overline{A}_{i+1}) = \dim(\overline{A}) - 1$ and $\dim(\overline{B}_{i+1}) = \dim(\overline{B}) - 1$, hence it holds that $\dim(\overline{A}_{i+1}) > \dim(\overline{B}_{i+1}) + \dim(\overline{C}_{i+1})$.

Algorithm Information Exchange will repeat the above until $\overline{B}_i = \overline{C}_i = 0$. At this point, $B_i = C_i$, so clients b and c can now be treated as a single client and $\dim(\overline{A}_i) = \dim(\overline{A}_0) - i$. The combined client transmits the remaining space in $\overline{A}_i$, requiring i transmissions, making the total number of transmissions equal to $\dim(\overline{A}_i) + i = \dim(\overline{A}_0) - i + i = \dim(\overline{A}_0)$, which achieves the lower bound. ■

Next we must consider the case where the triangle inequality in Equation 5.6 is true. Before we consider this case, we must present some lemmas about the dimensionality of subspaces as the algorithm proceeds:

Lemma 11 *If the condition* $\dim(\overline{A}_i) \leq \dim(\overline{B}_i) + \dim(\overline{C}_i)$ *holds, then* $\dim(\overline{A}_i) > 0$ *and* $\dim(\overline{B}_i) > 0$*, unless* $\dim(A_i) = \dim(B_i) = \dim(C_i) = 0$

Proof: Assume that $\dim(\overline{A}_i) = 0$. Then for the inequality to hold, $\dim(\overline{B}_i) +$

$\dim(\overline{C}_i) = 0$, so $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i) = 0$. Assume that $\dim(\overline{B}_i) = 0$. Then, for the inequality to hold $\dim(\overline{C}_i) \geq \dim(\overline{A}_i)$, which contradicts Equation 5.5 unless $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i) = 0$. ■

Lemma 12 *If the condition* $\dim(\overline{A}_i) \leq \dim(\overline{B}_i) + \dim(\overline{C}_i)$ *holds, then* $\dim(\overline{A}_{i+i}) + \dim(\overline{B}_{i+i}) + \dim(\overline{C}_{i+i}) = \dim(\overline{A}_i) + \dim(\overline{B}_i) + \dim(\overline{C}_i) - 2$ *and* $\dim(W_{i+1}) = \dim(W_i) + 2$.

Proof: From Lemma 11, we know $\dim(\overline{A}_i) > 0; \dim(\overline{B}_i) > 0$. As long as $\dim(\overline{A}_i) > 0; \dim(\overline{B}_i) > 0$, client c will transmit a packet p_i as defined in Equation 5.7, and $\overline{A}_{i+1}$ and $\overline{B}_{i+i}$ are defined as in Equations 5.8 and 5.9. Therefore,

$$\dim(\overline{A}_{i+i}) + \dim(\overline{B}_{i+i}) + \dim(\overline{C}_{i+i}) = \dim(\overline{A}_i) + \dim(\overline{B}_i) + \dim(\overline{C}_i) - 2$$

and $\dim(W_{i+1}) = \dim(W_i) + 2$.

■

Next we show that if triangle inequality in Equation 5.6 holds, then for $i = 1, 2, \ldots$ it will hold that $\dim(\overline{A}_i) \leq \dim(\overline{B}_i) + \dim(\overline{C}_i)$ until the algorithm hits one of two specific states. In these states, all clients either have 1 or 0 packets remaining. In the latter case, the algorithm is done where in the former, two transmissions remain, as shown in Lemma 14.

Lemma 13 *If Equation 5.6 holds, then Algorithm* Information Exchange *will maintain* $\dim(\overline{A}_i) \leq \dim(\overline{B}_i) + \dim(\overline{C}_i)$ *until* $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i) = d$ *where* $d = 1$ *or* $d = 0$.

Proof: Suppose that $\dim(\overline{A}_i) \leq \dim(\overline{B}_i) + \dim(\overline{C}_i)$ holds. After a transmission in round i, we have $\dim(\overline{A}_{i+1}) = \dim(\overline{A}_i) - 1$, $\dim(\overline{B}_{i+1}) = \dim(\overline{B}_i) - 1$, and $\dim(\overline{C}_{i+1}) = \dim(\overline{C}_i)$ since a single packet has been transmitted by client c to clients a

and b. If $\dim(\overline{A}_i) \neq \dim(\overline{C}_i)$, then 1 was subtracted from each side and the inequality $\dim(\overline{A}_{i+1}) \leq \dim(\overline{B}_{i+1}) + \dim(\overline{C}_{i+1})$ holds.

If $\dim(\overline{A}_i) = \dim(\overline{C}_i)$, this implies $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i)$ and it holds that $\dim(\overline{C}_{i+1}) > \dim(\overline{A}_{i+1})$ and $\dim(\overline{C}_{i+1}) > \dim(\overline{B}_{i+1})$. If we set $d = \dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i)$, the resulting inequality is $d \leq d - 1 + d - 1$, which is only false if $d \leq 1$. ■

Lemma 14 *Algorithm* Information Exchange *is optimal for* $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i) = 1$*, requiring* 2 *transmissions.*

Proof: If $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i) = 1$, then a trivial lower bound is 2 transmissions. A client must transmit in the next round, and that client cannot learn any new information. Therefore, that client needs at least one additional round to learn any new packets.

Algorithm Information Exchange will select the client with the greatest dimensionality to make a transmission. In this case, all clients have the same dimensionality, so without loss of generality, assume client c transmits. Then $\dim(\overline{C}_{i+1}) = 1$ and $\dim(\overline{A}_{i+1}) = \dim(\overline{B}_{i+1}) = 0$. At this point, clients a and b now contain all packets within their subspace since the dimensionality of $\overline{A}_{i+1}$ and $\overline{B}_{i+1}$ is 0, and client c is missing only one dimension since dimensionality of $\overline{C}_{i+1}$ is 1. The algorithm will pick either client a or client b to make a transmission to client c, transmitting the final dimension of $\overline{C}_{i+1}$. After 2 transmissions, all clients contain all packets, which achieves the lower bound. ■

Next we wish to prove how the algorithm behaves in the general case where triangle inequality in Equation 5.6 is true. There are two cases, depending on whether $\dim(N)$ is even or odd. If $\dim(N)$ is odd, the algorithm will continue until $\dim(\overline{A}) = \dim(\overline{B}) = \dim(\overline{C}) = 1$, as shown in Lemma 15. If $\dim(N)$ is even, then the algorithm will complete with $\dim(\overline{A}) = \dim(\overline{B}) = \dim(\overline{C}) = 0$, as shown in Lemma 16. Finally we show that in both of these cases, the number of transmissions is optimal in Lemma 17.

Lemma 15 *If Equation 5.6 is true and* $\dim(N)$ *is odd, then in execution of Algorithm* Information Exchange $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i) = 1$ *with* $i = \frac{\dim(N)-3}{2}$.

Proof: From Lemma 13, we know that inequality $\dim(\overline{A}_i) \leq \dim(\overline{B}_i) + \dim(\overline{C}_i)$ holds until $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i) = d; d = \{0, 1\}$. Furthermore from Lemma 12, we know that as long as Equation 5.6 holds, $\dim(\overline{A}_i) + \dim(\overline{B}_i) + \dim(\overline{C}_i) = \dim(N) - 2 * i$, so $\dim(\overline{A}_i) + \dim(\overline{B}_i) + \dim(\overline{C}_i)$ is odd for all i at least until $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i) = d; d = \{0, 1\}$. If $d = 1$, then the number of transmissions to this point is $\frac{\dim(N)-3}{2}$ from Lemma 12. Assume that the inequality $\dim(\overline{A}_i) \leq \dim(\overline{B}_i) + \dim(\overline{C}_i)$ holds until $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i) = 0$, then $\dim(\overline{A}_{i-1}) + \dim(\overline{B}_{i-1}) + \dim(\overline{C}_{i-1}) = 2$ since two sets must each be reduced in dimensionality by 1 as long as Equation 5.6 holds. Since 2 is even, this contradicts Lemma 12. ■

Lemma 16 *If Equation 5.6 is true and* $\dim(N)$ *is even, then in execution of Algorithm* Information Exchange $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i) = 0$ *for* $i = \frac{\dim(N)}{2}$.

Proof: From Lemma 13, we know that inequality $\dim(\overline{A}_i) \leq \dim(\overline{B}_i) + \dim(\overline{C}_i)$ holds until $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i) = d; d \in \{0, 1\}$. Thus, by Lemma 12, it holds that $\dim(\overline{A}_i) + \dim(\overline{B}_i) + \dim(\overline{C}_i) = \dim(N) - 2 * i$. This implies that $d \neq 1$. Therefore, Equation 5.6 holds until $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i) = 0$, which, by Lemma 12, occurs at iteration $i = \frac{\dim(N)}{2}$. ■

Lemma 17 *Algorithm* Information Exchange *is optimal for three clients when* $\dim(\overline{A}_0) \leq \dim(\overline{B}_0) + \dim(\overline{C}_0)$.

Proof: When $\dim(\overline{A}_0) \leq \dim(\overline{B}_0) + \dim(\overline{C}_0)$, one can define a lower bound of $\left\lceil \frac{\dim(N)}{2} \right\rceil$ since $\dim(N)$ packets must be shared and in any given round, and at most 2 clients can learn a single packet.

If $\dim(N)$ is even, we know from Lemma 16 that Algorithm Information Exchange takes $\frac{\dim(N)}{2}$ transmissions, which achieves the lower bound.

If $\dim(N)$ is odd, we know from Lemma 15 that Algorithm Information Exchange takes $\frac{\dim(N)-3}{2}$ transmissions to reach the point where $\dim(\overline{A}_i) = \dim(\overline{B}_i) = \dim(\overline{C}_i) = 1$, and from Lemma 14 requires two additional transmissions to complete. This totals $\frac{\dim(N)-3}{2} + 2 = \frac{\dim(N)+1}{2}$ transmissions. Since $\dim(N)$ is odd, the total transmissions made is $\left\lceil \frac{\dim(N)}{2} \right\rceil$, which achieves the lower bound. ■

Theorem 18 *Algorithm* Information Exchange *is optimal for three clients*

Proof: Follows from Lemmas 10, and 17. ■

2. Upper Bound

In the general case, clients will merge as their subspaces become identical. At some point, the algorithm will pass through a point where only three unique client subspaces exist.

Theorem 19 *With a network containing n packets, and an optimal packet exchange of* α *packets, Algorithm* Information Exchange *will transmit no more than* $\frac{19}{16}\alpha$ *packets.*

Proof: Define γ as the number of packets transmitted in Algorithm Information Exchange. Define n_0 as the lowest number of packets known by any client when the algorithm starts. Define n_1 as the lowest number of packets known by any client at the point when there are three unique subspaces. Define $\beta = n - n_1$.

In the case where Algorithm Information Exchange has three unique subspaces and the triangle inequality in Equation 5.6 is not met, then the client with the highest packet count has at least the same number of packets as every other client combined. Therefore, it, and other clients in the same set, must have been transmitting to reach this point. This would require $n_1 - n_0$ transmissions. From Lemma 10, this requires an additional $n - n_1$ transmissions to complete, totaling $n - n_0$ transmissions. This is a lower bound, so in this case it is optimal.

When Algorithm Information Exchange has three unique subspaces and Equation 5.6 is met, there have been at most $\frac{4}{3}(n_1 - n_0)$ transmissions, since there exists at least four clients prior to this point, and the client with the least number of packets receives at least three packets out of every four transmissions. From Lemma 17, this requires an additional $\frac{3}{2}$ transmissions to complete. The total packet count is

$$\gamma \leq \frac{4}{3}(n_1 - n_0) + \frac{3}{2}\beta = \frac{4}{3}(n - \beta - n_0) + \frac{3}{2}\beta = \frac{4}{3}(n - n_0) + \frac{1}{6}\beta.$$

Since Equation 5.6 is true, then $\alpha \geq \frac{3}{2}\beta$, so $\gamma \leq \frac{4}{3}\alpha + \frac{1}{4}\alpha = \frac{19}{16}\alpha$. ■

3. Example with 4 Clients

While the modified algorithm by Rouayheb et al. is optimal for 3 clients, it is not optimal for 4 clients. Consider the example in Figure 19. In this example, a client knows all packets below the line, while it wants all packets above the line.

Consider the case where client 1 transmits first (since it has the most number of packets), and chooses to transmit the packet $c + d$. This packet is beneficial to all

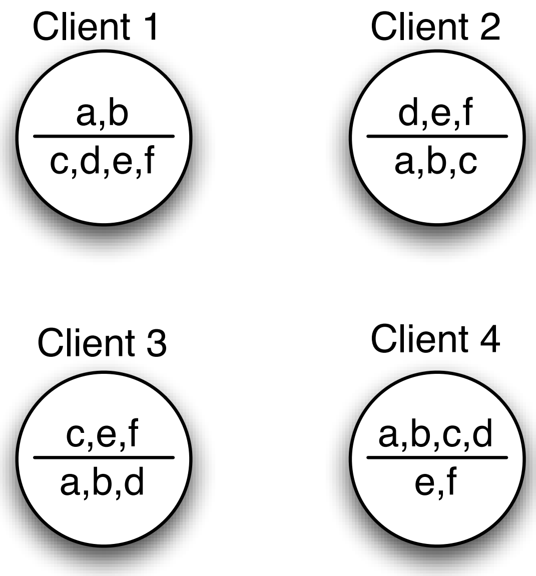


Fig. 19. Example Multicast Exchange With 4 Clients and 6 Packets.

other clients. With this new information, clients 2 and 3 are able to decode packets d and c respectively, making these two client posses identical information. Next, client 1 transmits packet $e + c + d$, which again is beneficial to everyone. At this point, the combined client 2 and 3 has the most information, so it chooses to transmit a, and since it has the same packet count as client 1, it still chooses to transmit and it sends b. Again, these two packets are beneficial to all other clients. Lastly, the combined client 2 and 3 needs packet f, and so it is transmitted by client 1. This results in a total of 5 transmissions.

Consider the schedule where client 2 transmits $a+c$, client 3 transmits $b+d$, then client 1 transmits $a+e$, followed by $b+f$. This schedule meets all the requirements for all clients in 4 transmissions. Additionally, if client 4 is removed from this problem, the algorithm by Rouayheb et al. without modification could behave in the same manner as it did with 4 clients. In this case, when client 4 is removed, packets e and

f are unique to client 1, which is why the modification to Algorithm Information Exchange is required.

It is important to note that with three clients, after the transmission of unique packets and the subsequent removal of knowledge common to all clients from the problem, every packet is known to exactly 2 clients. When the same conditions are imposed on 4 clients, packets are known to either 2 or 3 clients. This provides some intuition as to why the algorithm is optimal for 3 clients, but not necessarily optimal for 4. In addition, the sub-optimality of the transmissions in the above example is due to a poor transmission choice for the first packet rather than a poor choice in which client should transmit first.

D. A Randomized Algorithm

In this section, we present a randomized algorithm for the problem at hand. Our algorithm assumes a large finite field $\mathbb{F}$ of size q and provides an optimal solution with high probability. Later in the section will show that the size of the field can be reduced to $O(k)$, without increasing the total number of transmissions.

For clarity, we describe and analyze the behavior of the algorithm in terms of encoding vectors, rather than original packets. That is, instead of saying that a packet $p_i = \sum_{x_j \in X} \gamma_i^j x_j$ has been transmitted, we say that we transmit the corresponding encoding vector $\gamma_i = [\gamma_i^1, \gamma_i^2, \ldots, \gamma_i^n]$.

The algorithm operates in rounds. Assume that in round i, the encoding vector γ_i is transmitted by client c_j. Then, the transmitted vector γ_i is a random linear combination of the unit vectors in U_j, i.e., $\gamma_i^g = 0$ for $x_g \notin X_j$; other elements of γ_i are selected at random from the field $\mathbb{F}$. We denote $\Gamma_i = \{\gamma_1, \ldots, \gamma_i\}$ as the set of encoding vectors that have been transmitted up to and including round i.

The steps performed by the algorithm can be summarized as follows:

1. **For** $i \leftarrow 1, 2, \ldots, n$ **do**:

   (a) Select a client c_j for which the set $U_j \cup \Gamma_{i-1}$ is of maximum rank, i.e.,

   $$j = \arg\max_{c_j \in C}\{\text{rank}(U_j \cup \Gamma_{i-1})\};$$

   (b) Create a new encoding vector γ_i, such that $\gamma_i^g = 0$ for $x_g \notin X_j$, otherwise γ_i^g is a random element of field $\mathbb{F}$.

   (c) If for each $c_l \in C$ it holds that $\text{rank}(U_l \cup \Gamma_i) = n$, **go to** Step 2.

2. **Return** i encoding vectors $\{\gamma_1, \ldots, \gamma_i\}$

We proceed to analyze the correctness and optimality of the algorithm. Consider an iteration i of the algorithm. We denote OPT_i as the optimal number of packets that still need to be transmitted after round i,i.e. in addition to the first i transmissions, in order to satisfy the demands of all the clients.

Lemma 20 *With probability at least* $1 - \frac{n}{q}$, *it holds that* $OPT_i = OPT_{i-1} - 1$

Proof: Recall that the set $\Gamma_{i-1} = \{\gamma_1, \ldots, \gamma_{i-1}\}$ contains the packets that have been transmitted so far. Let Q_{i-1} be an optimal set of encoding vectors required to complete the information transfer. That is, Q_{i-1} includes OPT_{i-1} encoding vectors such that (i) each vector is a linear combination of U_l for some $c_l \in C$; (ii) for each client $c_l \in C$ it holds that the set $\Gamma_{i-1} \cup Q_{i-1} \cup U_l$ is of rank n.

Let $\mu = \text{rank}(U_j \cup \Gamma_{i-1})$ be the rank of the set of encoding vectors available to client c_j at the beginning of iteration i. We observe that OPT_{i-1} is at least $n - \mu + 1$. This follows from the fact that c_j is the client that has the maximum rank at the beginning of iteration i. If there exists a client with strictly lower rank than μ, then this client would require at least $n-(\mu-1)$ transmissions. Otherwise, if all clients have the same rank, then the required number of transmissions is also at least $n-\mu+1$ since the first client to transmit still needs $n-\mu$ transmissions to complete. This argument is similar to the lower bound in [8]. Thus, there exists at least one packet, v, that can be removed from Q_{i-1} to give $\tilde{Q}_{i-1} = Q_{i-1} \setminus \{v\}$ such that $\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_j$ remains of rank n.

Let c_l be a client in $C \setminus \{c_j\}$. We prove that with probability at least $1-\frac{n}{q}$ it holds that $\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_l \cup p_j$ is of rank n. Note that the rank of vector set $S_l = \Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_l$ is at least $n-1$. We only need to consider the case in which S_l is of rank $n-1$ since a client with S_l of rank n is unaffected by the removal of packet v. Let ζ_l be the normal vector to the span of S_l. In what follows, we show that ζ_l and γ_i are not orthogonal with high probability, i.e, the inner product $\langle \zeta_l, \gamma_i \rangle$ between ζ_l and γ_i is not equal to zero with probability at least $1-\frac{n}{q}$. This will suffice to prove the claim that $\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_l \cup p_j$ is of rank n with probability at most $1-\frac{n}{q}$.

Note that ζ_l can be written as

$$\zeta_l = \sum_{u_g \in U_j} \beta_g u_g + \sum_{u_g \in \overline{U}_j} \beta_g u_g,$$

where $\overline{U}_j$ is the set of unit encoding vectors that correspond to $\overline{X}_j = X \setminus X_j$.

We show that there exists $u_g \in U_j$ such that $\beta_g \neq 0$. If we suppose that it is not the case, then ζ_l can be expressed as $\zeta_l = \sum_{u_g \in \overline{U}_j} \beta_g u_g$. Note that for each $u_g \in \overline{U}_j$, the span of $\Gamma_{i-1} \cup \tilde{Q}_{i-1}$ must include a vector $v_g = u_g + \sum_{u_t \in U_j} \gamma_t u_t$. Since v_g is orthogonal to ζ_l, this implies that β_g is equal to zero for each $u_g \in \overline{U}_j$. This, contradicts the fact that ζ_l is not identical to zero.

Recall that γ_i is a random linear combination of vectors in U_j, i.e., $\gamma_i = \sum_{u_g \in U_j} \gamma_i^g u_g$ where γ_i^g are random coefficients over a field $\mathbb{F}$. Therefore, inner product $\langle \zeta_l, \gamma_i \rangle$ can be written as

$$\langle \zeta_l, \gamma_i \rangle = \sum_{u_g \in U_j} \beta_g \gamma_i^g.$$

Let $\hat{U}$ be a subset of U_j such that for each $u_g \in \hat{U}$ it holds that $\beta_g \neq 0$. We have showed above that the set $\hat{U}$ is not empty. Thus, $\langle \zeta_l, \gamma_i \rangle = \sum_{u_g \in \hat{U}} \beta_g \gamma_i^g$. Since for each $u_g \in \hat{U}$, γ_i^g is a random variable chosen independently of $\{\beta_g \; u_g \in \hat{U}\}$ the probability that $\langle \zeta_l, \gamma_i \rangle$ is equal to zero is at most $\frac{1}{q}$.

By using the union bound we can show that the probability that $\langle \zeta_l, \gamma_i \rangle = 0$ for some client $c_l \in C$ is bounded by $\frac{n}{q}$. Thus, with probability at least $1 - \frac{n}{q}$ for each client $c_l \in C$ it holds that $\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_l \cup p_j$ is of rank n. Therefore, after iteration i of the algorithm, the information transfer can be completed within $OPT_{i-1} - 1$ transmissions by using vectors in $\Gamma_{i-1} \cup \tilde{Q}_{i-1} \cup U_l$. ■

Theorem 21 *The algorithm computes, with probability at least $1 - \frac{n^2}{q}$, an optimal solution for the data exchange problem, provided that the size q is larger than n.*

Proof: Let OPT the be the optimum number of transmissions required to solve the information exchange problem. Note that $OPT_0 = OPT$. By Lemma 20, after each iteration, the number of required transmissions reduces by one with probability at least $(1-\frac{n}{q})$. Thus, the information transfer will be completed after OPT iterations with probability at least

$$(1 - \frac{n}{q})^{OPT} \geq (1 - \frac{n}{q})^n \geq 1 - \frac{n^2}{q},$$

where the last inequality holds for $q > n$. ■

By selecting a sufficiently large q (i.e., $q \geq 4n^2$), we can guarantee that the probability of success is at least $3/4$. Probability of success can be amplified to be

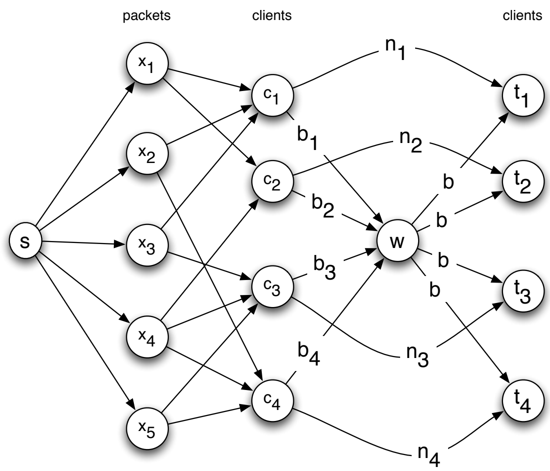


Fig. 20. Example Multicast Graph for 4 Clients and 5 Packets.

arbitrary close to 1 by performing multiple iterations and choosing the iteration that yields the minimum number of transmissions.

Corollary 22 *For any $\varepsilon > 0$ the algorithm can find an optimal solution to the information exchange problem with probability at least $1 - \varepsilon$ in time polynomial in the size of the input and $\log(\varepsilon)$.*

1. Reducing the Field Size

We can now construct a multicast problem as shown in Figure 20 to reduce the required field size to $|\mathbb{F}| \geq k$. The multicast setting consists a source node s and of 4 layers. The first layer has n nodes corresponding to n source packets. The source node s is connected by link to each node in layer 1. Layer 2 comprises k nodes corresponding to k clients. An existing edge e_{ij} between node i in the first layer and node j in the second layer means that client c_j knows packet x_i. Client nodes in layer 2 are connected to a single node, w, in layer 3, where the edge capacity b_j represents the number of transmissions from c_j determined by the algorithm. And finally, w distributes coded packets to all k destination client nodes with edge capacities equal to $b = \sum_{j=1}^{k} b_j$. Obviously, client c_j is interested in all n source packets but also has side information X_j, which can also be represented by direct edges from the second to the last layer with capacities equal to n_j. This is a standard multicast problem of transmitting n packets from the source node s to k destinations. Using the results in [57], we can find a network coding solution to the problem as long as $|\mathbb{F}| \geq k$.

We have thus shown that with linear coding we can achieve the optimal number of transmissions and achieve the capacity of the equivalent multicast problem. Hence, linear coding is sufficient for the information data exchange problem.

E. Conclusion

We presented a randomized algorithm that finds an optimal solution for the cooperative data exchange problem with high probability. While the algorithm gives a solution over a relatively large field, we showed that the field size can be reduced, through an efficient procedure, without any penalty in terms of the total number of transmissions. We also proved that the deterministic algorithm by Rouayheb et al. in [8] is optimal for three clients as well as a proof that this algorithm's chosen client transmission order is part of an optimal solution.

CHAPTER VI

CONCLUSION

This dissertation demonstrates that the reliability of a network contains many factors. Some networks need to be able to survive a natural disaster, such as a cellular network which needs to carry emergency calls, and others need to be able to survive random day to day failures. Even in the presence of failures, a sender can transmit its data in a manner so that clients can recover from limited failures. Clients may even work together to help each other recover data lost to some clients but retrieved by others.

In the analysis of the cellular infrastructure, this dissertation shows that the reliability can be evaluated using the combination of analysis of structural reliability in a hurricane, network and traffic modeling, and simulation. This technique provides a method for both assessing the reliability of the system as well as a framework to test alternative topologies. Through the addition of some increased redundancy, the reliability of the system can be significantly increased. Constructing a mesh network out of microwave dishes or adding a fiber link across the network's leaves showed a significant increase in the network's availability during in the aftermath of a hurricane.

When examining the simulation techniques for analysis of rare events, this dissertation presented two efficient methods for analysis of a network's expected loss of traffic (ELT). Compared to a Monte Carlo technique which was significantly accelerated through the use of a hash table, these techniques showed significant improvement in speed. The cut enumeration provided a fast lower bound on the amount of traffic lost, but it only counted cases where a network was disconnected. In cases where a few links are cut, and traffic is rerouted, network traffic lost due to congestion is missed by this technique. The used of genetic algorithms with niching provide a tighter bound while still presenting its results more quickly than the Monte Carlo technique. In cases of low probability, both of these techniques provided results which were also more accurate.

In the multicast problem, this dissertation provides a method for encoding parity packets which are effectively linearly independent of its source packets. This allows such parity packets to be transmitted in a coding network along with its source packets without concern of generating a packet which is identically 0. With this technique, one can establish a multicast connection to k clients which can withstand the failure of a single edge in the network without any action on the part of the source in response to the failure.

When clients in a multicast problem can communicate with one another in a broadcast channel, they are able to fill in gaps themselves if another client has the required data. This dissertation presents a technique where clients make random transmissions over a very large field which will be optimal with very high probability. Furthermore, after each client's transmission count is known, the field size can be reduced without affecting the transmission count. It also shows that the previously proposed algorithm is optimal for 3 clients, and that an optimal solution exists with the same transmission order proposed by this algorithm.

We rely on networks for our modern communications. Techniques, like those presented in this dissertation, can help to improve a network's reliability by providing the means to quickly analyze a complex system. In addition, providers can use techniques presented here to increase the reliability of their content delivery even when a network failure occurs during transmission. Also users can cooperate with one another to better their experience by fill in each other's gaps in shared content.

REFERENCES

[1] L.K. Comfort and T.W. Haase, "Communication, coherence, and collective action: The impact of Hurricane Katrina on communications infrastructure," *Public Works Management & Policy*, vol. 10, no. 4, pp. 328, 2006.

[2] K. Banipal, "Strategic approach to disaster management: Lessons learned from Hurricane Katrina," *Disaster Prevention and Management*, vol. 15, no. 3, pp. 484–494, 2006.

[3] W. Grover, *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking*, Upper Saddle River, NJ: Prentice Hall PTR, 2004.

[4] H. Zang and B. Mukherjee, "Connection management for survivable wavelength-routed WDM mesh networks," *SPIE Optical Networks Magazine*, vol. 2, no. 4, pp. 17–28, July 2001.

[5] A. Shahine, "Internet services in Egypt and India disrupted," http://in.reuters.com/article/topNews/idINIndia-31677820080130?sp=true, 2008, [Online; accessed 4-January-2010].

[6] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[7] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782 – 795, 2003.

[8] S. El Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," in *To Appear in Information Theory Workshop*, Cairo, Egypt, 2010.

[9] T. Rappaport, *Wireless Communications: Principles and Practice*, Upper Saddle River, NJ: Prentice Hall PTR, 2001.

[10] D. Tipper, C. Charnsripinyo, H. Shin, and T. Dahlberg, "Survivability analysis for mobile cellular networks," in *Proceedings Communication Networks and Distributed Systems Modeling and Simulation Conference 2002*, San Antonio, TX, 2002, pp. 27–31.

[11] A.P. Snow, U. Varshney, and A.D. Malloy, "Reliability and survivability of wireless and mobile networks," *Computer*, vol. 33, no. 7, pp. 49–55, 2000.

[12] J.R. Goodman, M.D. Vanderbilt, and M.E. Criswell, "Reliability-based design of wood transmission line structures," *Journal of Structural Engineering*, vol. 109, no. 3, pp. 690–704, 1983.

[13] R.E. Melchers, *Structural Reliability Analysis and Prediction*, New York: Wiley, 1999.

[14] HJ Dagher, "Reliability of poles in NESC grade C construction," in *Rural Electric Power Conference, 2001*, Little Rock, AK, 2001, pp. C4/1–C4/6.

[15] Z. Huang, D.V. Rosowsky, and P.R. Sparks, "Hurricane simulation techniques for the evaluation of wind-speeds and expected insurance losses," *Journal of Wind Engineering & Industrial Aerodynamics*, vol. 89, no. 7-8, pp. 605–617, 2001.

[16] H. Liu, R.A. Davidson, D.V. Rosowsky, and J.R. Stedinger, "Negative binomial regression of electric power outages in hurricanes," *Journal of Infrastructure Systems*, vol. 11, pp. 258–267, 2005.

[17] S.R.H.S.D. Guikema, S.M.Q.K.H. Lee, and D.R.R.A. Davidson, "Estimating the spatial distribution of power outages during hurricanes in the Gulf Coast region," *Reliability Engineering & System Safety*, vol. 94, no. 2, pp. 199–210, 2009.

[18] P.N. Georgiou, "Design wind speeds in tropical cyclone-prone regions.," Ph.D. dissertation, University of Western Ontario, 1987.

[19] P.J. Vickery and L.A. Twisdale, "Wind-field and filling models for hurricane wind-speed predictions," *Journal of Structural Engineering*, vol. 121, no. 11, pp. 1700–1709, 1995.

[20] D. Rosowsky, P. Sparks, and Z. Huang, "Wind field modeling and hurricane hazard analysis," *South Carolina Sea Grant Consortium and Civil Engineering Dept., Clemson Univ., Clemson, SC*, 1999.

[21] H. Liu, "Electric power distribution system performance in Carolina hurricanes," *Natural Hazards Review*, vol. 4, pp. 36–45, 2003.

[22] J. G. Johnston, "Mesopolis: A virtual city for water distribution system research applications, procedural documentation," Department of Electrical and Civil Engineering, Texas A&M University, available from https://ceprofs.civil.tamu.edu/kbrumbelow/Mesopolis/Mesopolis procedural documentation.pdf, [Online; accessed 24-May-2008].

[23] "OPNET Modeler. OPNET Technologies, Inc.," http://www.opnet.com, [Online; accessed 4-Feb-2008].

[24] C. J. Colbourn, *The Combinatorics of Network Reliability*, New York: Oxford University Press, Inc., 1987.

[25] R. Karp and M. Luby, "Monte-Carlo algorithms for the planar multiterminal network reliability problem," *Journal of Complexity*, vol. 1, pp. 45–64, 1985.

[26] R. M. Kapp, M. Luby, and N. Madras, "Monte-Carlo approximation algorithms for enumeration problems," *Journal of Algorithms*, vol. 10, no. 3, pp. 429–448, 1989.

[27] A. Konak and A.E. Smith, "Network reliability optimization," in *Handbook of Optimization in Telecommunications*, M. G. C. Resende and P. M. Pardalos, Eds. New York: Spinger Science + Business Media, 2006.

[28] E. Manzi, M. Labbe, G. Latouche, and F. Maffioli, "Fishman's sampling plan for computing network reliability," *Reliability, IEEE Transactions on*, vol. 50, no. 1, pp. 41–46, Mar 2001.

[29] Y. Liu and K. S. Trivedi, "A general framework for network survivability quantification," in *Proceedings of GI/ITG Conference on Measuring, Modelling, and Evaluation of Computer and Communication Systems (MMB)*, Dresden, Germany, 2004, pp. 369–378.

[30] N. Samaan and C. Singh, "Adequacy assessment of power system generation using a modified simple genetic algorithm," *IEEE Transactions on Power Systems*, vol. 17, no. 4, pp. 974–981, Nov 2002.

[31] N. Samaan and C. Singh, "A new method for composite system annualized reliability indices based on genetic algorithms," in *Proceedings of IEEE Power Engineering Society Summer Meeting, 2002*, Chicago, IL, 25-25 July 2002, vol. 2, pp. 850–855.

[32] C. Singh and L. Wang, "Role of artificial intelligence in the reliability evaluation

of electric power systems," in *Proceedings of IEEE International Conference on Electrical and Electronics Engineering (ELECO07)*, Bursa, Turkey, 2007, pp. 1–8.

[33] C. Singh and L. Wang, "An alternative to Monte Carlo simulation for system reliability evaluation: Search based on artificial intelligence," in *Proceedings of International Conference on Reliability and Safety*, Udaipur, India, 2007.

[34] D. Karger, "A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem," *SIAM Journal on Computing*, vol. 29, no. 2, pp. 492–514, 1999.

[35] L. Jereb, T. Jakab, and F. Unghváry, "Availability analysis of multi-layer optical networks," *Optical Networks Magazine*, vol. 3, no. 2, pp. 303–314, March/April 2002.

[36] L. Jereb, "Network reliability: Models, measures and analysis," in *Workshop on Performance Modelling and Evaluation of ATM Networks*, Ilkley, UK, 1998, pp. T02/1–T02/10.

[37] J. Levendovszky, L. Jereb, Z. Elek, and G. Vesztergombi, "Adaptive statistical algorithms in network reliability analysis," *Performance Evaluation*, vol. 48, no. 1-4, pp. 225–236, 2002.

[38] B. Dengiz, F. Altiparmak, and A.E. Smith, "Efficient optimization of all-terminal reliable networks, using an evolutionary approach," *IEEE Transactions on Reliability*, vol. 46, no. 1, pp. 18–26, Mar 1997.

[39] A. A. Akyamac, J. F. Labourdette, Z. Bogdanowicz K. Hua, S. Chaudhuri, and S. French, "Optical mesh network modeling: Simulation and analysis of

restoration performance," in *Proceedings of the 18th Annual National Fiber Optic Engineers Conference*, Dallas, TX, 2002, pp. 1408–1418.

[40] M. Menth, R. Martin, and J. Charzinski, "Capacity overprovisioning for networks with resilience requirements," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 87–98, 2006.

[41] M. Clouqueur and W.D. Grover, "Availability analysis of span-restorable mesh networks," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 810–821, May 2002.

[42] Z. Havic, "Availability level of optical cables within unique communication system," *18th International Conference on Applied Electromagnetics and Communications (ICECom 2005)*, pp. 1–4, Oct. 2005.

[43] D.O. Lawrence G.H. Zeidler, P.R. Bark and O.I. Szentesi, "Reliability of fiber optic cable systems," in *Proceedings of International Conference on Communications (ICC'80)*, Seattle, WA, June 1980, pp. 10.3.1–10.3.4.

[44] J. Suurballe and R. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, pp. 325–336, 1984.

[45] V. V. Vazirani, *Approximation Algorithms*, New York: Springer, March 2004.

[46] V. V. Vazirani and M. Yannakakis, "Suboptimal cuts: Their enumeration, weight and number," in *ICALP '92: Proceedings of the 19th International Colloquium on Automata, Languages and Programming*, London, UK, 1992, pp. 366–377, Springer-Verlag.

[47] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[48] D. E. Goldberg and J. Richardson, “Genetic algorithms with sharing for multimodal function optimization,” in *Proceedings of the 2nd International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, Hillsdale, NJ, 1987, pp. 41–49.

[49] M. Scheffel, “Adaptation of failure scenario based resilience schemes toward availability guarantees,” *Journal of Optical Networking*, vol. 5, no. 7, pp. 521–531, 2006.

[50] W. D. Grover, *Mesh-Based Survivable Transport Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking*, New York: Prentice-Hall, 2003.

[51] E. Ayanoglu, C.-L. I., R. D. Gitlin, and J.E. Mazo, “Diversity coding for transparent self-healing and fault-tolerant communication networks,” *IEEE Transactions on Communications*, vol. 41,, no. 11, pp. 1677–1686, 1993.

[52] T. Ho, M. Médard, and R. Koetter, “An information-theoretic view of network management,” *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1295–1312, April 2005.

[53] D. S. Lun, M. Médard, R. Koetter, and M. Effros, “Further results on coding for reliable communication over packet networks,” in *IEEE International Symposium on Information Theory (ISIT 05)*, Adelaide, Australia, 2005, pp. 1848–1852.

[54] D. S. Lun, M. Médard, and M. Effros, “On coding for reliable communication over packet networks,” in *Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, 2004, pp. 3–20.

[55] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371 – 381, 2003.

[56] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proceedings of the IEEE International Symposium on Information Theory*, Yokohama, Japan, 2003, pp. 442–447.

[57] S. Jaggi, P. Sanders, P.A. Chou, M. Effros, S. Egner, K. Jain, and L.M.G.M. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, June 2005.

[58] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros, "Resilient network coding in the presence of byzantine adversaries," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2596–2603, June 2008.

[59] D. Silva, F.R. Kschischang, and R. Koetter, "A rank-metric approach to error control in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 3951–3967, Sept. 2008.

[60] R. Koetter and F.R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3579–3591, Aug. 2008.

[61] T. Ho and D. S. Lun, *Network Coding: An Introduction.*, Cambrige, UK: Cambridge University Press, 2008.

[62] C. Fragouli and E. Soljanin, *Network Coding Fundamentals (Foundations and Trends in Networking)*, Hanover, MA: Now Publishers Inc, 2007.

[63] R. Yeung, *Information Theory and Network Coding*, Cambridge, MA: Springer, 2008.

[64] P.A. Chou and Yunnan Wu, "Network coding for the internet and wireless networks," *Signal Processing Magazine, IEEE*, vol. 24, no. 5, pp. 77–85, Sept. 2007.

[65] C. Gkantsidis and P.R. Rodriguez, "Network coding for large scale content distribution," *Proceedings IEEE INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, pp. 2235–2245 vol. 4, March 2005.

[66] E. Gabidulin, "Theory of codes with maximum rank distance," *Problems of Information Transmission*, vol. 21, no. 1, pp. 3–14, July 1985.

[67] D. Silva and F. R. Kschischang, "Security for wiretap networks via rank-metric codes," *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, pp. 176–180, july 2008.

[68] S. El Rouayheb, A. Sprintson, and C. Georghiades., "Simple network codes for instantaneous recovery from edge failures in unicast connections," in *UCSD Workshop on Information Theory and its Applications*, San Diego, California, February 2006.

[69] Y. Birk and T. Kol, "Coding-on-demand by an informed source (ISCOD) for efficient broadcast of different supplemental data to caching clients," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2825–2830, June 2006.

[70] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Ko, "Index coding with side information," in *Proc. of the 47th Annual IEEE Symposium on Foundations of*

Computer Science (FOCS), Berkeley, CA, 2006, pp. 197–206.

[71] S. El Rouayheb, A. Sprintson, and C. N. Georghiades, "On the relation between the index coding and the network coding problems," in *Proc. of IEEE International Symposium on Information Theory (ISIT08)*, Toronto, Ontario, Canada, 2008, pp. 1823–1827.

[72] D. Shah, *Gossip Algorithms (Foundations and Trends in Networking)*, vol. 3, Hanover, MA: Now Publishers Inc, 2007.

[73] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity–Part I: System description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, Nov. 2003.

VITA

Graham Broadt Booker received his Bachelor of Science degree in Electrical Engineering from Texas A&M University in 2002. He entered the Electrical Engineering graduate program in June of 2002 and entered the Ph.D. program in Computer Engineering in September of 2003. His research interests include network reliability, network coding, and computer security. He has published papers in these topics and plans to continue work in industry.

Mr. Booker can be reached at 214 Zachry Engineering Center, TAMU 3128, College Station, TX 77843-3128. His email address is gbooker@tamu.edu.