FINDING AND MITIGATING GEOGRAPHIC VULNERABILITIES IN MISSION CRITICAL MULTI-LAYER NETWORKS

A Dissertation
IN
Telecommunications and Computer Networking
and
Electrical and Computer Engineering

Presented to the Faculty of the University
of Missouri–Kansas City in partial fulfillment of
the requirements for the degree

DOCTOR OF PHILOSOPHY

by
MICHAEL TODD GARDNER

M. S., University of Kansas, Lawrence, USA, 2002
B. S., University of Missouri, Columbia, USA, 1990

Kansas City, Missouri
2016

FINDING AND MITIGATING GEOGRAPHIC VULNERABILITIES IN MISSION CRITICAL MULTI-LAYER NETWORKS

Michael Todd Gardner, Candidate for the Doctor of Philosophy Degree

University of Missouri–Kansas City, 2016

ABSTRACT

In Air Traffic Control (ATC), communications outages may lead to immediate loss of communications or radar contact with aircraft. In the short term, there may be safety related issues as important services including power systems, ATC, or communications for first responders during a disaster may be out of service. Significant financial damage from airline delays and cancellations may occur in the long term. This highlights the different types of impact that may occur after a disaster or other geographic event. The question is *How do we evaluate and improve the ability of a mission-critical network to perform its mission during geographically correlated failures?*

To answer this question, we consider several large and small networks, including a multi-layer ATC Service Oriented Architecture (SOA) network known as SWIM. This research presents a number of tools to analyze and mitigate both long and short term geographic vulnerabilities in mission critical networks. To provide context for the tools, a

disaster planning approach is presented that focuses on Resiliency Evaluation, Provisioning Demands, Topology Design, and Mitigation of Vulnerabilities.

In the Resilience Evaluation, we propose a novel metric known as the Network Impact Resilience (NIR) metric and a reduced state based algorithm to compute the NIR known as the Self-Pruning Network State Generation (SP-NSG) algorithm. These tools not only evaluate the resiliency of a network with a variety of possible network tests, but they also identify geographic vulnerabilities.

Related to the Demand Provisioning and Mitigation of Vulnerabilities, we present methods that focus on provisioning in preparation for rerouting of demands immediately following an event based on Service Level Agreements (SLA) and *fast* rerouting of demands around geographic vulnerabilities using Multi-Topology Routing (MTR). The Topology Design area focuses on adding nodes to improve topologies to be more resistant to geographic vulnerabilities.

Additionally, a set of network performance tools are proposed for use with mission critical networks that can model at least up to 2nd order network delay statistics. The first is an extension of the Queueing Network Analyzer (QNA) to model multi-layer networks (and specifically SOA networks). The second is a network decomposition tool based on Linear Algebraic Queueing Theory (LAQT). This is one of the first extensive uses of LAQT for network modeling. Benefits, results, and limitations of both methods are described.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Graduate Studies, have examined a dissertation titled "Finding and Mitigating Geographic Vulnerabilities in Mission Critical Multi-Layer Networks," presented by Michael Todd Gardner, candidate for the Doctor of Philosophy degree, and hereby certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Cory Beard, Ph.D., Committee Chair
Department of Computer Science & Electrical Engineering

Reza Derakhshani, Ph.D., Co-Discipline Chair
Department of Computer Science & Electrical Engineering

Deep Medhi, Ph.D.
Department of Computer Science & Electrical Engineering

Appie Van de Liefvoort, Ph.D.
Department of Computer Science & Electrical Engineering

Yugyung Lee, Ph.D.
Department of Computer Science & Electrical Engineering

Ken Mitchell, Ph.D.
Department of Computer Science & Electrical Engineering

James P. G. Sterbenz, Ph.D.
Information and Telecommunications Technology Center
The University of Kansas

CONTENTS

ILLUSTRATIONS

TABLES

ACKNOWLEDGEMENTS

I would like to begin by thanking my research advisor, Dr. Cory Beard. As a non-traditional student, he has been infinitely patient as I continually tried to share my academic life with my work life and my home life. Dr. Beard always encouraged me to follow my passion and that has enabled me to live in the world of highly available mission critical networks, a world that I share with my employer, the Federal Aviation Adminstration (FAA).

My co-discipline advisor, Dr. Reza Derakhshani showed me that the tools you use are only limited by your ability to use those tools. I feel like that many of the problems that the world faces today (including some of the problems discussed in my research) are simply problems of complexity that will eventually will be solved using tools like Machine Learning, Neural Networks, and Swarm Optimization.

One of the things I have enjoyed the most during my time at UMKC is collaboration with other professors and students. Dr. Deep Medhi's energy as it relates to all things networking enabled us to make interesting contributions related to routing, network optimization, and resilience metrics. I think his ability to provide insight into any networking problem no matter how obscure is his superpower. Dr. James Sterbenz's passion for resilient networks has shown us that these problems can be solved and resilient networks can be built regardless of the challenges. I would like to thank Dr. Sterbenz for allowing us to work with the Resilinets Group at the University of Kansas.

Dr. Appie Van de Liefvoort spent endless hours with me as I slowly learned the

art that is Linear Algebraic Queueing Theory (LAQT). After finally understanding the power of this incredible set of tools, I will continually be an evangelist for these ideas and concepts.

I would like to thank my committee members, Dr. Medhi, Dr. Van de Liefvoort, Dr. Ken Mitchell, Dr. Yugi Lee, and Dr. Sterbenz for their continual support, guidance, and encouragement as I have worked through my research.

I would also like to thank several folks at FAA, is where my passion for mission critical networks begins. Maureen Cedro has always been a champion for me and my efforts to study interesting problems during my graduate work. She along with Dr. Steve Bradford helped create the research project that is the subject of much of this work. Dr. Ed Zakrewski, Dr. Dave Garbin, Dr. Krishnan Belakrishnan, and many others associated with the FAA have inspired me continually with new ideas, concepts, about the mission critical world in which we all live. I also cannot forget Tom Duncan, who at the last minute helped me to introduce the problem in a way that everyone would understand and appreciate.

Finally, I would like to thank my family. They are my rock. My wonderful wife, Jill has always been supportive during all of my academic endevours. My kids, Nick and Beckett have never really known me when I wasn't involved in some sort of research project.

CHAPTER 1

INTRODUCTION

The Federal Aviation Administration (FAA) operates a large and complex network to support air traffic control (ATC) operations [44]. The network topology reflects the hierarchical configuration of ATC facilities as aircraft progress through the different phases of flight. Each class of ATC facility covers a specific airspace defined by the combination of a geographical footprint and an altitude range above that geographical footprint. The basic categories of ATC facilities actively involved in the control of aircraft are listed in Table 1 [113].

Table 1: ATC Facilities [113]

Facility	Phase of Flight
Air Traffic Control Tower (ATCT)	Takeoff & Landing
Terminal Radar Approach Control (TRACON)	Airport to approximately 50 Miles from airport
Air Route Traffic Control Center (ARTCC)	TRACON to TRACON (Enroute)

The ATC facilities require telecommunications connectivity to remote air/ground radio sites for voice communication with aircraft, surveillance radars, weather radars, navigational aids, and adjacent ATC facilities for coordination of hand-offs as aircraft enter and leave their airspace [44].

To support these communications requirements, the FAA operates a highly available, diverse, and telecommunications network that leverages the infrastructure of commercial carriers throughout the United States and abroad. Major ATC facilities are designed with diverse access paths with sufficient capacity to support the re-routing of all critical services onto the alternate path providing high availability for those services [44]. *But what happens when both paths are lost simultaneously?* While rare, this type of *6-sigma event* have occurred.

In 2007, the Memphis ARTCC experienced an outage when a circuit card at an AT&T Central Office failed causing a failure on an AT&T fiber ring network that had been considered to be fault tolerant [39, 150]. A portion of the radar services and air-to-ground frequencies was lost. The outage lasted for approximately three hours. 566 delays were caused by the outage [150]. At the time of outage, 220 airplanes were in the Memphis airspace. Controllers used cell phones to establish communications with adjacent facilities. Airplanes with working air-to-ground communications relayed information via radio from controllers to airplanes without working air-to-ground communication channels. Ultimately, the airplanes were landed safely and the airspace was closed [150].

More recently in September 2014, a disgruntled contractor set fire to the room housing the communications network infrastructure at the Chicago ARTCC and attempted to sever the communications lines. The facility had to be evacuated, including the air traffic controllers, and the damage to the facility was so extensive that it took three weeks for it to re-open. While the FAA's telecommunications network was robust enough to re-route

traffic around the Chicago ARTCC, the FAA faced the challenge of redirecting connections between Chicago and its remote sites to adjacent ARTCCs. This was achieved over the course of several days and ATC operations were eventually restored to full capacity by temporarily locating the air traffic controllers at the adjacent facilities [22, 72].

Like Memphis, there were two types of impacts. First, there were immediate safety related impacts. In this case, 40 aircraft were in the air over the airspace when radio and radar contact was lost [72]. Controllers used cell phones to notify adjacent facilities to attempt to make contact with airplanes. Airplanes also used emergency procedures (like the 121.5 MHz emergency frequency) to attempt to re-establish contact with air traffic control [72]. All aircraft landed safely.

The longer term impacts were related less to safety and more to damage. During the following two weeks, thousands of flights were canceled and an estimated $350 Million was lost by the airline industry [71]. In 2014, the average cost to the airline industry was $82 per airline minute of delay not counting the cost to consumers in wasted time and productivity [4]. A large and lengthy delay causing outage can cause staggering financial damages.

These examples demonstrate an interesting pattern of events during outages on mission critical networks. First, there are normally safety critical services that are needed immediately following an event either because the event interrupted mission operations or because the safety critical services are needed to manage the event. This would be the case for First Responders during a disaster. Second, outages on mission critical services may have very high impacts over time until full restoration is achieved. This would apply

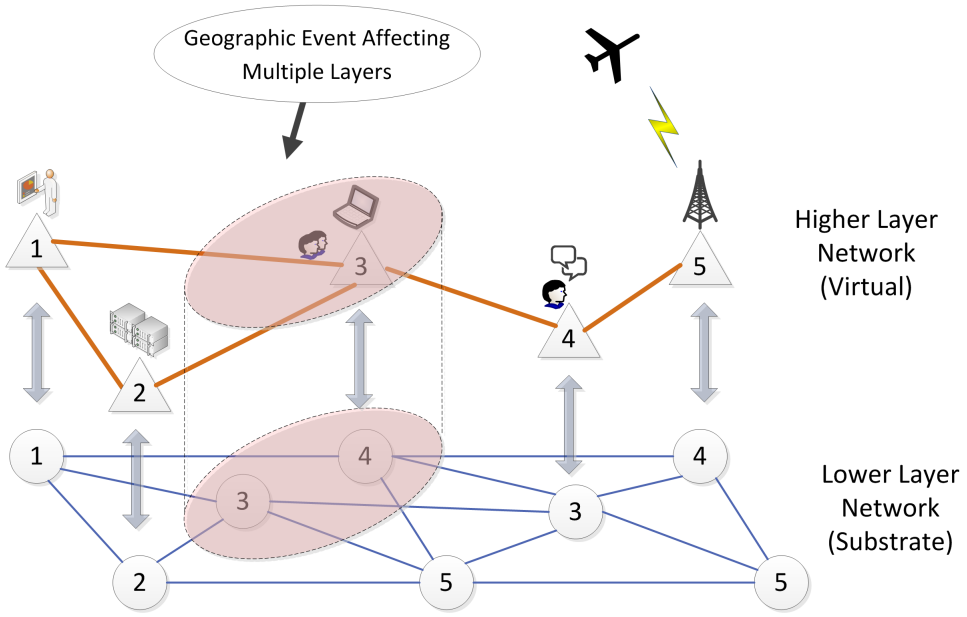


Figure 1: Layered Network showing Geographic Failure

to services like air traffic control, power systems, and other critical infrastructure.

During disasters, the criticality of mission critical network services may even increase as these services are called upon to assist with search, rescue, and recovery as well as provide critical infrastructure such as power or Air Traffic Control (ATC) services during search and rescue as is discussed in [157] and [144]. Given these two examples, the question becomes:

How do we evaluate and improve the ability of a Mission-Critical Network to perform its mission during geographically correlated failures?

Multilayer networks warrant additional concern since geographic events can affect nodes and infrastructure in multiple layers of the network simultaneously (see Fig. 1)

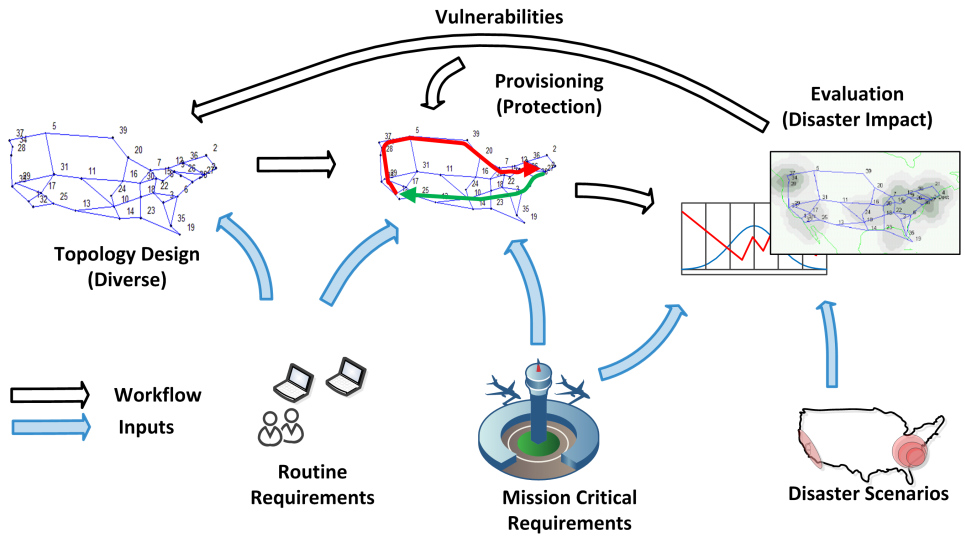


Figure 2: Disaster Planning Workflow and Approach

leading to challenges in determining network performance of higher layer services. The Federal Aviation Administration (FAA) System Wide Information Management (SWIM) network is an example of a mission critical multi-layer network [143] that was studied during this research. Chapter 2 describes the FAA SWIM Network.

1.1 Disaster Planning for Mission Critical Services

Research related to geographic vulnerabilities in networks frequently focuses on one aspect of the problem like topology considerations or graph theoretic network measures. The users (especially mission critical users) and the *impact* that a geographic event will have on those users is rarely considered. To address these challenges, we propose a *Disaster Planning Approach* that includes vulnerability analysis, topology improvement, provisioning, restoration, and routing improvement in the context of disasters.

Mature resilience frameworks like the Resilinets framework [147] provide a roadmap

for the design of resilient networks in the face of network challenges such as disasters. The active phase referenced in the Resilinets framework is *defend, detect, remediate, recover*, also known as the D^2R^2 phase of the framework. The R^2 of the framework refers to *remediate* and *recover*, which are similar to the short term and long term impacts described.

In Figure 2, we see a proposed design process for a mission critical network. The inputs to the process include routine demands, mission critical demands, and disaster scenarios. The disaster planning approach presented includes the following components:

- Topology Design
- Provisioning Demands
- Resiliency Evaluation
- Mitigation of Vulnerabilities

The workflow generally begins with a topology selection constrained with demand endpoints. We then proceed to provisioning demands on the topology. Disaster evaluation that uses potential scenarios follows. This yields a set of vulnerabilities. With the new vulnerabilities, topology selection and/or provisioning is revisted and augmentation with new nodes or links can be made. To properly frame the conversation about geographic vulnerabilities in networks, we define the following terms that will be used throughout this work.

DEFINITION 1.1. *Geographic Vulnerability is the geographic area of a network that if attacked can cause significant impact to the function of the entire network.*

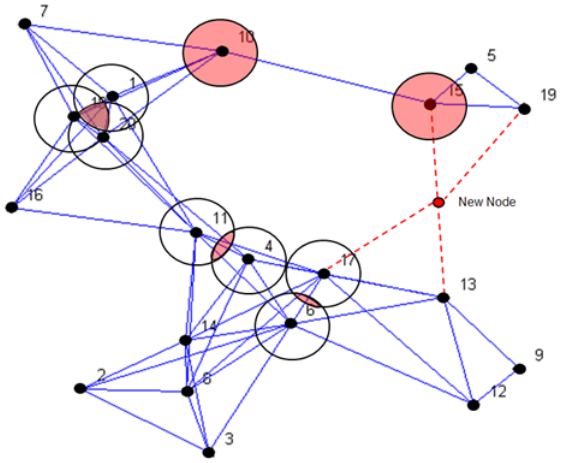


Figure 3: Wireless Network showing Geographic Vulnerabilities and a New Node Location that Eliminates the Vulnerabilities

DEFINITION 1.2. *A Geospatial Event is an event that can cause a geographic vulnerability.*

DEFINITION 1.3. *The Threat Radius is the physical radius of a geospatial event that is used to define the geographic vulnerability.*

In this work, it is usually assumed that during a geospatial event, all nodes in the threat radius will fail.

1.1.1 Topology Design

Topology Design is related to the layout and connectivity of the physical infrastructure. A topology's inherit relationship to geography can present vulnerabilities that can only be mitigated by augmentation of the topology as opposed to having additional capacity in the network. This is illustrated in Figure 3 where a new node was added to

eliminate the vulnerabilities located at nodes 10 and 15.

Goals of topology design can be to minimize costs by optimizing node/edge locations or to provide coverage that is resistant to failures (random or correlated). Frequently, topology design is based on reachability constraints (or diverse reachability constraints). Heuristic and optimization based methods are used to for topology design purposes.

1.1.2 Demand Provisioning

Demand provisioning is generally accomplished via heuristics (like Find First (FF) algorithms) or via optimization algorithms. Demand provisioning can take into account path diversity and redundancy. Two formulations are generally used for demand provisioning that consider path diversity. The first is the Path-Link Formulation [121], which relies on path generation algorithms as an input to the formulation. This is useful if the path generation is designed for a specific purpose. For example, two paths may be physically separated by a certain distance or avoid a certain area. This is explored in Chapter 7

The second formulation is the Node-Link Formulation [121], which generates traffic flow paths within the formulation. This is useful because it avoid the path generation problem. There are methods within Node-Link formulations to create two paths that avoid each other. Incorporating geographic distance between paths is more challenging and is the subject of Chapter 6.

1.1.3 Resiliency Evaluation

Resiliency evaluation as it relates to geographically correlated failures and as it is used in this work involves two concepts:

1. Locating geographic areas that present vulnerabilities to a network
2. Creating a metric that evaluates the capability of a network to deal with geographically correlated failures given the mission of the network.

Resiliency evaluation is traditionally performed using a variety of methods which include simulation, state space analysis, graph theoretic methods, and other techniques. Some methods take into account the mission (or demands) on the network. Other methods focus on reachability to locate vulnerabilities. Metrics are sometimes used to quantify resiliency. Some graph theoretic metrics include betweeness, algebraic connectivity, node degree, and clustering coefficient. If the mission of the network is considered, performance metrics must be computed as a part of the resiliency evaluation.

The mission of the network is considered in the Resilinets framework as the relationship of *Network State* to the *Service State* [145]. Additionally, in [52], Garbin and Shortle discuss the concept of resilience curves which connect the number of failed components to the network function.

Performability is one of the metrics that considers the network mission during the analysis. From Medhi [101], performability uses the probability of network events and the network performance associated with that event to calculate an average network performance. This is shown in (1.1), where S_i is the system state, $X[S_i]$ is the network

measure at state S_i, and $Pr[S_i]$ is the probability of network state S_i. N is the number of network elements.

The problem with this approach is that high probability events tend to dominate performability and thus the design of the network. If the network is relatively reliable, the performance during the time when the network is *up* will dominate the performability calculation. With mission critical networks, we are primarily interested in survivability and resilience rather than performability alone. To address this concern, we explore the concept of network impact as opposed to network measure.

$$P = \sum_{i=1}^{2^N} Pr\left[S_i\right] X\left[S_i\right] \tag{1.1}$$

1.1.4 Vulnerability Mitigation

As discussed earlier, vulnerability mitigation may have safety related short term impacts and longer term impacts that tend to be related to cost. Different vulnerability mitigation techniques can be used for each of these impacts.

Geodiverse routing techniques can be used to mitigate short term impacts [32]. Other concepts include coordinated switching techniques taking advantage of the speed of lower layer switching versus upper layer switching. Geographically correlated events present considerable challenges for routing algorithms. Route flapping occurs during large outages delaying the time to route convergence and potentially leading to route instability [16].

Longer term impacts tend to me addressed with capacity planning techniques. The

concern with many approaches is the lack of attention paid to planning for the event itself and the mission critical services that will need to be restored. As is pointed out in Section 1, the criticality of services during a geographically correlated event may even change due to the event itself.

1.2 Performance in Mission Critical Networks

Is the mission met with this set of services operating at this level of service? The key to this question is understanding the mission and relationship of the network to that mission. Evaluating the resiliency of a network necessitates understanding the impact on the mission of the network. If we make the following assumptions:

ASSUMPTION 1. *Unacceptable high latency and/or packet loss by a service is considered an outage of the service.*

This assumption implies that the PDF of the latency or packet loss directly affects the availability of the service. First order (means based) network analysis techniques to determine latency are common. Jackson Networks [75] are a well known, first order, open queueing network solution. Techniques described in [105] also provide basic first order response time analysis for web services. If second order statistics are desired, Queueing Network Analyzer (QNA) [155] is a tool that provides these capabilities.

Multi-layer networks present additional challenges. Layered Queueing Networks (LQN) are one of the techniques used to solve these challenges. LQNs are solved frequently using means based analysis or simulation. Most multi-layer network performance that is tied to critical services is simply modeled using Discrete Event Simulation (DES).

These techniques all have disadvantages that hinder their adoption in multi-layer mission critical network performance analysis. Jackson networks and other first order analysis techniques are not accurate enough since they tend to make assumptions that performance is based on exponential distributions. QNA had not been used extensively for multi-layer modeling. This is one of the approaches proposed in Chapter 3. LQNs are typically solved analytically using means based approaches and have the same concerns as other first order approaches. Simulation is generally intractable for large networks and has the other disadvantage that rare events are difficult to simulate without specifically simulating the event itself.

Menasce [105] and [104], addressed web service QoS specifically. Generally, request/reply is the service type discussed here. The important characteristics discussed include response time, availability, reliability, predictability, security, and throughput. The Publish/Subscribe message pattern is also important to the Air Traffic Control (ATC) environment.

In the ATC environment, certain types of data are more critical to receive than other data types. In addition, timeliness is more important to certain types of data. For example, aircraft position data to controllers may have much tighter availability and timeliness constraints than perhaps aircraft position data to a web site that tracks aircraft for display on the world wide web, which in fact may need to be deliberately delayed for security reasons. This is also illustrated in the example in [104].

- Availability
- Survivability

- Response Time/Latency
- Message Loss Rate/Error Rate
- Security/Trust

It is interesting that a trade-off tends to exist between many of these qualities. For example, if buffer size is increased or message persistence is allowed, message loss can be minimized but at the expense of response time or latency. If availability is defined by the percentage of the time that messages arrive on time, availability would increase as the latency metrics increase. This work looks at the first three items primarily, leaving security and trust issues for separate work.

1.3 Service Level Agreements for Mission Critical Applications

During significant geographically correlated events, like earthquakes and hurricanes, a couple of challenges exist. The first challenge is to ensure that *mission critical* services are restored first and other services follow as efficiently and quickly as possible. The other challenge is ensuring that the network was provisioned properly prior to the event to facilitate restoration of *mission critical* services during a variety of different potential events. Do the network paths exist to route around the event? And is there enough capacity to handle the additional traffic?

Service Level Agreements (SLA) may provide much of the information needed to answer both of these challenges. SLAs can contain information about the importance of the service as well as the necessary service level objectives (SLO) for the service. We can

both provision the network with the information in the SLAs as well as restore the most important services based on this information.

Even though there is a large body of work related to assembling selecting services for web service composition based on QoS parameters in SLAs, we have not seen work that attempts to provision services or demands across a network using SLAs that have the ability to provide geographic diversity in the network.

In mission critical networks, the network typically has a strong role to play due to the high availability and sometimes low latency requirements that are imposed by mission critical services. In [11], Badidi describes a brokered SLA service for SOA that could be used by mission critical networks to provide and implement QoS. The author provides an extensive discussion about how to implement service level objectives (SLO). One of the advantages of brokered SLA services is that during events, the network has the ability to use the SLA to restore the most important services.

In [15], methods of restoration using SLAs during failures are discussed. The goal here is to ensure services are restored at a minimum QoS level during the event. We try to account for this by provisioning using SLA QoS parameters.

1.4 Contributions

This work addresses many of the components of the disaster planning approach described in Section 1 and shown in Fig. 2.

1.4.1 Performance Analysis of Mission Critical Multi-Layer Networks

Mission critical services frequently have tighter availability, latency, and packet loss requirements than is typically provided with traditional network analysis that relies on exponential assumptions for arrival and service processes. Services over multi-layer network infrastructure are also complex to analyze using traditional analysis techniques. Two network performance models and a novel random number generator are proposed.

The first performance model that is proposed was an extension of the classic Queuing Network Analyzer (QNA) by Ward Whitt [155] to include multi-layer analysis typical of a Service Oriented Architecture (SOA). Given the constraints of QNA (only includes 1st and 2nd order analysis), we proposed a linear algebraic queueing method that utilizes matrix exponential techniques. One of the significant advantages of these methods is the ability to use non-exponential arrival and departure processes in a network analysis tool. Additionally, simulation models to test the performance models yielded a novel matrix exponential random number generator. This results of this work are published at the IEEE MASCOTS and SCS SpringSim conferences [54, 57, 58]

1.4.2 Evaluation of Geographically Correlated Vulnerabilities in Multi-Layer Networks

A novel *Impact* based resilience metric is proposed known as the Network Impact Resilience (NIR) metric. In a manner similar to performability, it is based on a mapping of failure impact to network states. The Self-Pruning Network State Generation (SP-NSG) algorithm is created as a companion to the NIR. The SP-NSG is a reduced state space method of analyzing impact for important states. The NIR using the SP-NSG produces

two outputs. The first output is a metric that gives an indication of the disaster resilience of the network being evaluated. The second output is a list of specific failure modes or geographic vulnerabilities found. This work is published in the Journal of Network and System Management (JONS) [63] and at IEEE DRCN and IEEE CQR conferences [53, 56, 61].

1.4.3 Provisioning and Restoral of Mission Critical Services for Disaster Resilience

Provisioning ILPs and associated heuristics are proposed to solve the provisioning problem considering geographic vulnerabilities. This work was a collaboration with the University of Kansas and their diverse path algorithm was used with this work (see Section 1.5). Restoration algorithms and ILPs are presented to provide a *priority* of restoration based on service survivability parameters. Finally, Service Level Agreement (SLA) extensions are proposed to communicate the various mission critical parameters. The results of this work were recently published at the IEEE DRCN conference [59].

1.4.4 Topology Improvements to Avoid High Impact Geographic Events

Heuristic based Integer Linear Programs (ILP) are proposed here that augment network topologies by adding nodes to reduce geographic vulnerabilities found using the SP-NSG. Novel geographically based weighting algorithms are used with shortest path first algorithms to find diverse point-to-point paths as well as diverse point-to-many trees. Additionally, a novel *swarm optimization* approach is proposed that uses the SP-NSG natively. Each method has benefits. The results of this work were published at the IEEE Cogsima Conference [55].

1.4.5 Routing of Mission Critical Services during Disasters

Routing improvement during disasters is one of the vulnerability mitigation techniques considered for this work. Geographic Multi-Topology Routing (MTR) is proposed as a novel method to allow for *fast* switching to a stable routing topology during large and potentially cascading failures that may cause flapping or other routing pathologies in the base topology. In a manner similar to Section 1.4.4, geographically based weighting is used to divert the trunks of routing trees from vulnerable areas. This work is published in the Computer Networks Journal [62] and at IEEE DRCN Conference [60].

1.5 Collaborations and Other Contributions

The motivation for this research stems from two areas. First, this work is supported by a cooperative agreement between UMKC and the Federal Aviation Administration[1] to determine the suitability of a SOA for mission critical applications. The main question asked by this research agreement is: *Can SOA be used for mission critical systems?* Which suggests the following question: *Is the SOA resilient to disasters and other rare events?* Dr. Cory Beard is the Primary Investigator for this work.

Second, there is an overriding consensus that massive correlated failures in networks are not well understood. NSF has chosen to dedicate resources to a more comprehensive look at massive network failures in the form of a grant[2] to UMKC and Kansas University which also supports this work. Dr. Deep Medhi and Dr. James Sterbenz

[1]FAA-UMKC Cooperative Agreement - 11-G-018, "Analysis of the FAA NextGen SWIM Architecture". The Primary Investigator is Cory Beard.

[2]National Science Foundation grant No. CNS-1217736, " Resilient Network Design for Massive Failures and Attacks". The Primary Investigators are Deep Medhi and James Sterbenz.

are the Primary Investigators for this work. Additional publications related to this work include [31], [32].

1.6 Organization

The dissertation is organized as follows:

- Chapter 2 - SWIM Network - A Motivating Example
- Chapter 3 - Performance Analysis of Mission Critical Multi-Layer Networks
- Chapter 4 - Evaluation of Geographically Correlated Vulnerabilities in Multi-Layer Networks
- Chapter 5 - Provisioning and Restoral of Mission Critical Services for Disaster Resilience
- Chapter 6 - Topology Improvements to Avoid High Impact Geographic Events
- Chapter 7 - Routing of Mission Critical Services during Disasters
- Chapter 8 - Conclusions

CHAPTER 2

SWIM NETWORK - AIR TRAFFIC CONTROL EXAMPLE

The purpose of the Federal Aviation Administration (FAA) System Wide Information Management (SWIM) network [143] is to provide an open and flexible information management architecture that facilitates sharing of operational data among National Airspace System (NAS) entities in a secure and manageable fashion [149] Flight and traffic flow management (TFM), aeronautical information, and weather data can easily be shared with both internal and external users of SWIM. SWIM is based on a Service Oriented Architecture (SOA) multi-layer network concept.

A cooperative agreement between UMKC and the Federal Aviation Administration[1] asked the question:

Can SOA be used for mission critical systems?

Note that a significant part of this question would be:

Can the SOA provide the required Quality of Service (QoS)?

Can the SOA provide the required service availability?

This leads to the related question:

Is the SOA resilient to disasters and other rare events?

There is an overriding consensus that massive correlated failures in networks are

[1]FAA-UMKC Cooperative Agreement - 11-G-018, "Analysis of the FAA NextGen SWIM Architecture". The Primary Investigator is Cory Beard.

not well understood. This is unfortunately demonstrated by the communications experience during numerous natural and man-made disasters. This research project proposes models to asses QoS, availability, and network resilience of SOA networks.

SOA loosely binds applications to users using a common messaging format. Web Services generally extends SOA to include a publish/subscribe paradigm. In addition to publishing, the services are registered in a web services directory system that allows users to discover the service and subscribe. A typical publish/subscribe message distribution is shown in Figure 4. Web Service protocols govern the creation, description, publishing, discovery, invocation, and un-publishing of web services. SWIM implements Web Services using Simple Object Access Protocol/Extendable Markup Language (SOAP/XML) messaging to deliver data to NAS users. SWIM also implements a NAS service directory system that uses the Web Service Description Language (WSDL) to describe the services available to the users.

Message routing and delivery is handled by message brokers in the web service environment. From an architecture perspective, message brokers operate at the HTTP layer and above. This presents challenges to building QoS models because there is queuing of messages at the Web Service Application layer as well as the TCP and IP layers. From an availability perspective, hardware and software faults can occur at all three of those layers.

Evaluation of mission impact provided by a given set of network provided services over a particular network configuration is the most important metric of the performance of the network. In [105], Menasce clarified the need for a set of metrics at the web service

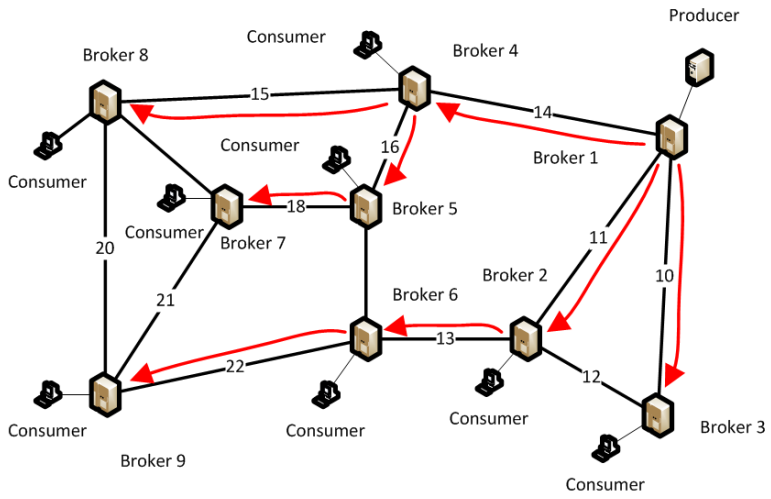


Figure 4: Typical Pub/Sub Message Distribution

layer that includes response time, availability, reliability, predictability, and cost. What is missing here is the answer to the question: *Is the mission met with this set of services operating at this level?* The key to the this question is understanding the connection between physical layer impacts to the network due to a disaster scenario and mission impact due to the subsequent degradation of the network provided services. Service Oriented Architecture (SOA) is one of the important mechanisms by which services are provided to a system over an underlying network.

2.1 Unique Challenges for ATC/SWIM

Due to the nature of ATC, SWIM is generally a highly *governed* network. The network provides security, QoS monitoring, etc... There is a large number of publishers and consumers. Generally the relationship is *not* one-to-one. It is one-to-many, with

multiple consumers being associated with each publisher and multiple publishers being used by each consumer. An example is text based weather data (ASOS). A given ASOS sensor located at an airport may publish its weather observations to multiple users at ARTCCs and TRACONs. And each ARTCC would receive multiple ASOS sensor data.

Finally, in SWIM there is more of an *abstraction* between consumers and producers than in a traditional web services relationship. In a traditional web services relationship where a customer may make a purchase from a vendor, the relationship between producer and consumer is one-to-one during the interaction with almost no involvement from the network (other than carrying data). In the SWIM/ATC environment, the network is involved in more aspects like governance, security, outage reporting, etc. In the SWIM/ATC environment, the consumers do not even need to know what producer is publishing the data.

Couple this environment with the basic question *How are extremely high availability SOA services specified?* and we have an interesting challenge. If we assume a *network of brokers* approach referred to here as the SWIM Enterprise Service Bus (ESB), we have the following context:

- ESB will need to *know* requirements
- ESB may make provisioning decisions based on the SLAs
- Some consumers may require high availability while others do not for the same data.
- Some consumers may require diversity and/or survivability

This seems to suggest that a strong centralized ESB is appropriate. Where the publisher and subscriber agree on an SLA that the ESB negotiates and enforces. One additional consideration is shown in Figure 5. The SWIM ESB needs to be interoperable with multiple ESBs and domains, including an *airborn* domain.

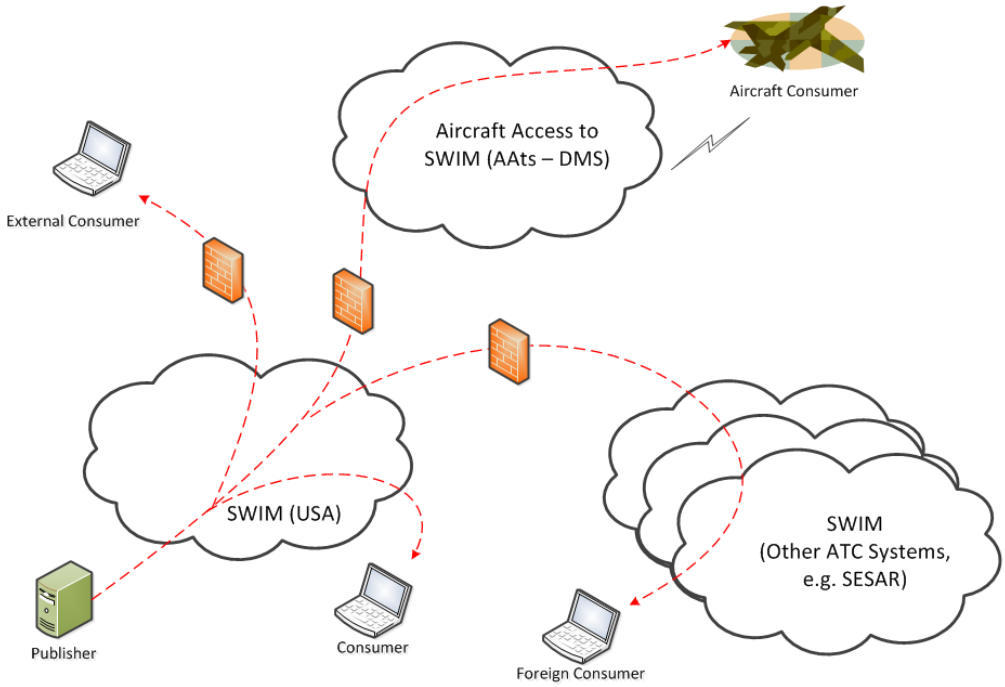


Figure 5: SWIM Interoperability

2.1.1 MC-SOA SLA Agreement Approaches

The two basic agreement styles referenced in [17] are Publisher Offered - Subscriber Requested (PO-SR) and Publisher Offered - Subscriber Requested - Middleware Confirmed (PO-SR-MC). Figures 6 and 7 illustrate possible SLA interactions based on

these agreement styles.

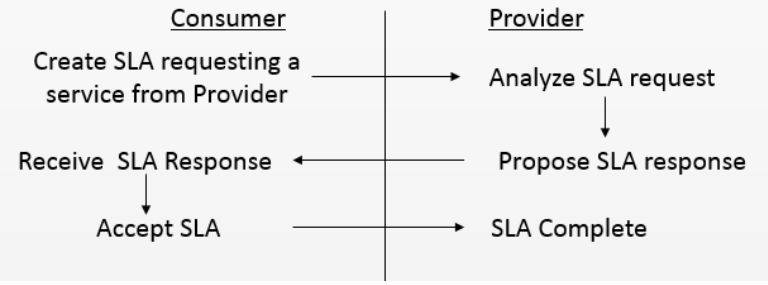


Figure 6: SLA based on PO-SR

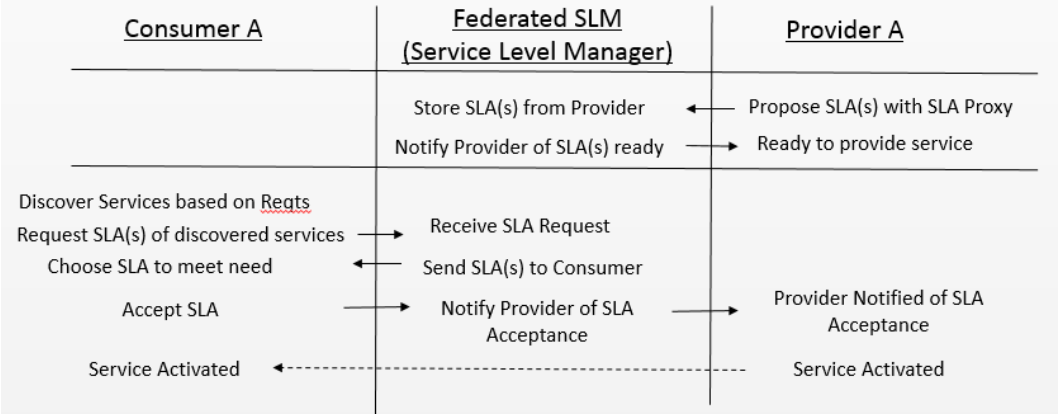


Figure 7: SLA based on PO-SR-MC

Based on these interactions, we compare the two approaches.

- PO-SR
  - Not scalable with multiple producers/consumers
  - SLA(s) have no central control, could be arbitrarily specified and enforced

- SOA would not be able to differentiate service

- PO-SR-MC

  - Enforcement of SLA(s) can be controlled by the ESB

  - Abstraction between producers and consumers is achieved

  - [11] and [27] both describe how a brokered SLA service might be implemented

The PO-SR-MC agreement style seems to provide the most advantages for a MC-SOA environment. In [20], some basic service level objectives (SLO) of a SOA SLA are described:

- Measurable Qualities: Accuracy, Availability, Capacity, Cost, Latency, Provisioning-related time, Reliable messaging, Scalability.

- Unmeasurable Qualities: Interoperability, Modifiability, Security

The language used for SLA(s) should be a common, flexible, supportable language. Main SLA descriptions languages include WSDL, WS-Agreement, WSLA, WS-Policy, others. As long as FAA Service Level Objectives (SLO) can be specified in an SLA, several languages would work.

A minimum recommended set of possible SLO(s) based on the QoS parameters in Section 1.2 include:

- Availability (multiple availabilities offered per product)

- Diversity (yes/no)

- Response Time/Latency (Provider + ESB + Consumer)
- Message Loss Rate (Maximum Rate)
- Error Rate (Maximum Rate)
- Security/Trust

Figure 8 shows an example of a service that is delivered as a high and low availability service to different consumers. Based on the SLA(s) negotiated, the SWIM could implement the service in the network based on the SLA. Notice that not only is availability selectable based on the products offered, diversity in the network is also selectable.

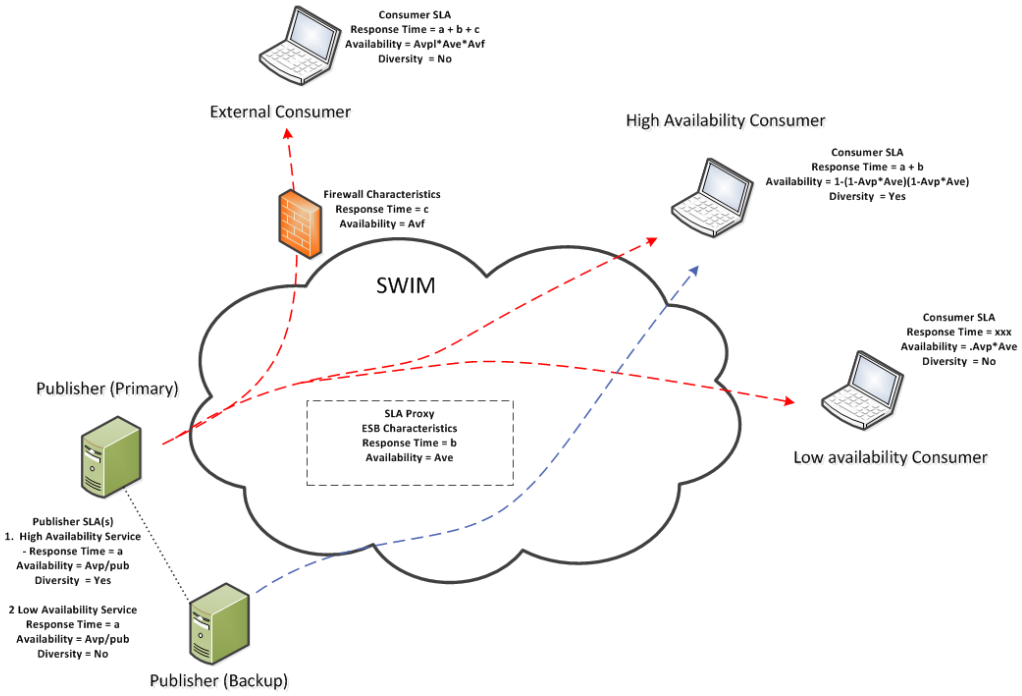


Figure 8: Example of High and Low Availability Products

CHAPTER 3

PERFORMANCE ANALYSIS OF MISSION CRITICAL MULTI-LAYER NETWORKS

In order to *evaluate* mission critical networks for geographic vulnerabilities, we must be able to determine if the network is performing the mission critical function with respect to the most important services. One of the challenges is that many mission critical applications tend to have requirements like high availability and/or low latency requiring accurate performance models.

In this chapter, the performance models used to study multi-layer networks are described. These include an extension of the well known Queueing Network Analyzer (QNA) and a network analysis tool based on Linear Algebraic Queueing Theory (LAQT). These new models are were published in [54] and [58].

Building network performance models that are capable of accurately predicting response time distributions for mission critical applications has not generally been possible for large networks. This is especially true for SOA Publish-Subscribe networks. There are a several reasons for the shortfall. Probability distributions representing the arrival and service processes are generally too complex to accurately represent in a large model without making the model intractable or using simplifying assumptions that do not capture important characteristics of the distribution, like the length of the tail. Jackson networks and Queueing Network Analyzer are examples of such models. Simulation models have difficulty generating rare events characterized by long tail distributions and

also have computational issues with large complex networks.

This is particularly important when flows with significantly different service times are mixed. Additionally, the Publish-Subscribe (Pub/Sub) messaging pattern creates a tree-like distribution where interior nodes may copy a message and send it out on multiple ports. A typical Pub/Sub distribution is shown in Figure 9. Many queueing models assume that splitting is done probabilistically and not by duplicating jobs (or messages).

The motivation for this work is to have the capability to create performance models for mission critical Service Oriented Architecture (SOA) systems like the Federal Aviation Administration (FAA) System Wide Information Management (SWIM) program [143]. Sachs et al. [135] report on observed message sizes and arrival rates associated with SOA systems. SOA adds complexity to an already complicated tri-modal packet size distribution on the internet as documented by Fraleigh in [48]. Moment matching [151] can be used on empirical samples to approximate complex traffic with probability distributions like Matrix Exponential (ME) distributions.

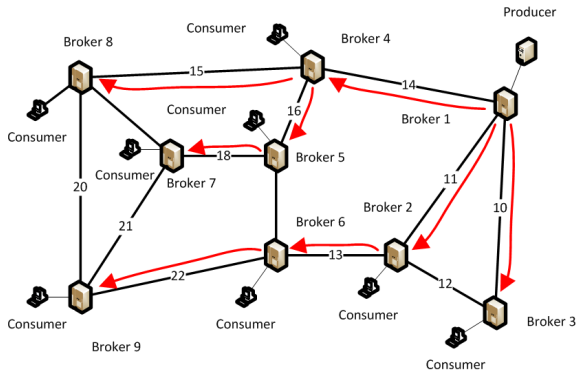


Figure 9: Typical Pub/Sub message distribution.

Matrix Exponential (ME) distributions used in linear algebraic queueing theory

(LAQT) methods are becoming more prevalent as system modelers are able to accurately incorporate various system conditions, frequently based on traffic samples. Since analytic methods to represent complex probability distributions are not tractable mathematically, we use ME probability distributions to approximate arrival and service distributions accurately. The only limitation on the types of distributions that can be represented is the size of the matrices needed to represent the distribution. Since LAQT models are *exact* in many operations, they make great models for research, enabling analytical results for problems that previously could not be solved analytically.

In this work, we use LAQT techniques in conjunction with a network decomposition approach to compute the resultant process of these three basic operations: superposition, splitting, departure. LAQT methods that compute the waiting time distribution across a single server are used to compute approximations for the response time distribution. Since state space explosion can occur with multiple operations across multiple nodes, moment matching techniques are used to limit state space. The advantage of this approach is that end-to-end network performance can be approximated with a minimum of parameters at any point in the network. Renewal approximations are still assumed due to the state space required to maintain correlation information. Research is on-going to eliminate the need for the renewal approximation. LAQT methods are sometimes viewed as complex when compared with traditional techniques. There are two good reasons to consider these techniques. First, LAQT models are matrix analytic extensions of familiar Markovian approaches, simplifying the LAQT operations used. Second, the extensive capabilities of LAQT models are worth the increase in model complexity. One of the

capabilities that is being explored in current research is the inclusion of correlation information in traffic flows from node to node. This would potentially eliminate many of the sources of error in network models.

Classic examples of network models that make simplifying assumptions to reduce complexity include Jackson networks [75] and the Queueing Network Analyzer (QNA) [155]. The Jackson network makes exponential assumptions about the arrival and service process. QNA was able to describe arrival and service processes using two parameters, mean and squared coefficient of variation (SCV) and then use linear equations to approximate these parameters at each node in the network. QNA achieved a complete two parameter approximation for network performance using *linear* equations making it the standard for network decomposition methods. The reasons the QNA model was chosen in this work as the baseline model over other models are:

- The model is tractable and multiple message classes (products) are possible. It is conceivable that a large SOA network may have hundreds of message classes each with its own SLA and arrival/service traffic distribution (which may not be exponential).

- The publish/subscribe and request/reply message patterns are handled easily with the customer creation feature of QNA.

- Multilayer modeling including fragmentation can also be handled using customer creation.

One of the features that make QNA an important tool for Pub/Sub system modeling is the

customer creation feature that is imbedded in QNA. This feature allows jobs (or in this case messages) to be duplicated or combined at a node. This was proposed as a possible solution to model Pub/Sub messaging environments in [54]. It is also possible to use this feature for multi-layer modeling, incorporating network layer fragmentation.

Our goal is to produce results that are more accurate than is possible with previous network decomposition approaches eventually creating a new class of network decomposition models using the powerful LAQT toolset. Additionally, the ability of the LAQT models to evolve to include new features like correlation both from superposed traffic streams and from tandem servers plays an important role. The organization of this paper is as follows. Related work is presented next. Then the LAQT and modified QNA models are described. Finally, the simulation model is described and results with all three models are presented.

3.1 Related Work

There are several threads of literature discussed in this section. First, works that specifically attempt to model SOA nodes using a variety of methods are discussed. Second, queueing network decomposition approaches are discussed. Next, LAQT techniques are reviewed. Our work lies in the intersection of these areas of research.

In [105], Menasce covers the traditional methods of analyzing the performance of SOA networks. These are based primarily on first order models with exponential assumptions. Badidi, Esmahi, and Serhani [12] presented a queueing model that allows for different classes of service to be directed to different queues in order to maintain the

SLAs of each traffic class. Both of these models are first moment models and therefore make significant simplifying assumptions. We are proposing methods here that do not require those assumptions. Kounev in [79] used a novel queueing Petri Nets approach to model the interactions at a SOA node. While this model seems to accurately reflect the interactions at a node, it does not seem to allow for widely varying message arrival and departure distributions.

Multiple works have used network decomposition approaches to predict network performance. Probably the most well known approach is the Queueing Network Analyzer (QNA). QNA was proposed by W. Whitt [155] in 1983 to address congestion in queuing networks. The product that resulted from this work provided a framework to estimate the 1st and 2nd moments of many network parameters including end-to-end delay. The authors in [155] and [154] point out the limitations to the assumptions that were made. The focus of our work is to create a model that improves on the limitations of the QNA model in the SOA Pub/Sub environment. In [122] Pongthawornkamol, Nahrst-edt, and Wang use concepts from QNA to improve their pub/sub model to include brokers modeled as Gi/G/1 queues.

There are a couple of works that share goals of our work. In [136] Sadre and Haverkort presents the FiFiQueue network analysis tool. FiFiQueue extended QNA, using a two parameter traffic descriptor and QNA assumptions for superposition and splitting. For the departure process, they use a phase type (PH/PH/1 or PH/PH/1/K) analysis. Our work does not extend QNA, instead it replaces the analysis for superposition, departure, and splitting with LAQT methods. Kim et al. [78] extends QNA with CorrQNA by

replacing the QNA squared coefficient of variation (SCV) linear equations with a regression equation that includes correlation information. Heindl et al. has proposed network decomposition methods that are based on LAQT methods in [70]. In addition, they used a Markovian Arrival Process (MAP) and moment matching methods that enable them to capture correlation induced in upstream queues. In [73], network decomposition is achieved using MAP processes as inputs and moments as inputs and outputs. This is done to reduce the state space requirements in large networks.

Several queueing models in a manner similar to Heindl that are significantly more complex have been designed with approximations that include correlation information. Mitchell and Van de Liefvoort [111] were able to construct a feed forward network using LAQT techniques that included correlated traffic inputs. It did not include superposition or splitting. However, in [10] and [77], the authors used three parameter traffic descriptions to build a simple queueing network with correlated inputs. It is still an open question whether these methods can be applied to large networks with multiple diverse inputs. In [126], Rabta proposed a QNA-like decomposition approach with an augmentation that incorporates simulations for components that are challenging to model analytically.

3.2 Queueing Network Analyzer for SOA (SOA-QNA)

For brevity, most of the details related to the inner workings of QNA are left to the reader in [155]. In [69], Haverkort also provides a good treatment of QNA. QNA generally assumes that there three basic operations in the queuing network. These are superposition, service, and splitting of renewal traffic streams. 1st and 2nd moment traffic

equations are formed and solved which yields the network node steady state statistics. From that, we can estimate the end-to-end delay statistics of the individual streams. The Queueing Network Analyzer (QNA) [155] uses a set of well known approximations for superposition, splitting, departure operations and waiting time calculations.

Algorithm 1 SOA-QNA Model

1: Assign node number to each message node and communications link.
2: From traffic route tables, find paths of each message type to each subscriber. Every pair represents a traffic class. Incorporate Message Exchange Pattern (MEP) information.
3: Determine mean service times and C^2s for each traffic class at each node. Collect the arrival statistics for each traffic class.
4: Convert traffic data into QNA *standard* input.
5: If chunking/multiplexing is assumed on the links, estimate the mean service time and C^2s of each traffic class.
6: Calculate the customer creation constant (γ) for each node. This is the departure rate leaving the node divided by the arrival rate.
7: Calculate 1st and 2nd order traffic equations and solve.
8: Calculate steady state network node statistics.
9: Calculate the end-to-end delay statistics for each traffic class

The model for superposition uses an approximation by Albin [6]. (3.1) shows the calculation of the SCV of the superposed process which is a convex combination of the weighted sum of the SCV of the input processes and the exponential, which has an SCV of one. The weighting function uses utilization and the arrival rates as inputs and can be found in [155]. The SCV for the ith arrival process is c_i^2 and c_H^2 is the SCV of the superposed process.

$$c_H^2 = w \sum \left(\frac{\lambda_i}{\sum_k \lambda_k} \right) c_i^2 + 1 - w \tag{3.1}$$

Since random splitting of a renewal process is also renewal, the splitting operations is easily handled by (3.2) [155]. c^2 is the SCV for the input arrival process and c_i^2 is the SCV of the ith split process with a probability of p_i.

$$c_i^2 = p_i c^2 + 1 - p_i \tag{3.2}$$

The work by Marshall [95] is used to create the linear departure approximation. It is then simplified based on the waiting time approximation shown in (3.4) and analysis by Whitt [155] to what we see in (3.3). c_s^2 is the SCV for the service process; c_a^2 is the SCV for the arrival process; and c_d^2 is the SCV of the departure process.

$$c_d^2 = \rho^2 c_s^2 + \left(1 - \rho^2\right) c_a^2 \tag{3.3}$$

Waiting time is approximated using the well known approximation for $Gi/G/1$ queues shown in (3.4) [80]. ρ is utilization; τ is the mean service time; and EW is the resultant expected (mean) waiting time. g is the Kramer and Langenbach-Belz approximation [80] used here when the SCV of the service process is greater than one, else it is set to 1.

$$EW = \tau\rho \left(c_s^2 + c_a^2\right) g/2 \left(1 - \rho\right) \tag{3.4}$$

QNA also has a convenient feature known as *customer creation* which uses a multiplicative factor at each node to create new customers after service. This is an extremely useful feature for modeling SOA message distribution to multiple nodes as was proposed in [54]. Our work borrows this concept and implements it using an LAQT method. The

reader is referred to [155] for more detailed information on the implementation of QNA.

3.2.1 Message Exchange Patterns (MEP) Implementation using QNA

The publish/subscribe MEP is implemented by estimating the distribution trees of each message and coding these distribution trees into the QNA route matrix (q_{ij}) and probability matrix (p_{ij}). This will place traffic on the appropriate nodes and links to account for the single source/multiple destination nature of pub/sub. γ is computed from the resultant traffic streams as shown in (3.5). k represents incoming links and m represents outgoing links.

$$\gamma = \frac{\sum_k \lambda_{k,out}}{\sum_m \lambda_{m,in}} \tag{3.5}$$

The request/response MEP is implemented in much the same manner. The request traffic flows and the resultant reply traffic flows are estimated independently and coded into the QNA route matrices. The arrival distribution of the request flow could influence the arrival distribution of the response flow. Incorporating the correlation between the two flows is still a subject of future research. Hybrid MEP, like those that involve a pub/sub notification and subsequent request/reply for file retrieval is implemented in a similar manner.

3.2.2 Message Chunking and IP Fragmentation

Modeling message chunking and link multiplexing involves modifying a couple of parameters. First, the service time τ_s for that product on the link needs to be adjusted for the message chunk size. Equations (3.6) - (3.10) show the service time model that was

used for links with chunking/multiplexing. We assumed a batch of MTU size packets and a single packet with a uniform distribution between MTU and zero. Next the customer creation factor (γ) must be set to accommodate for the additional packets on the link. (3.5) shows the calculation of the customer creation factor.

$$N = \frac{MsgSize}{ChnkSize} \tag{3.6}$$

$$x_s = \frac{8(ChnkSize)}{LinkSpeed} \tag{3.7}$$

$$OV_s = \frac{8(Overhead)}{LinkSpeed} \tag{3.8}$$

$$\overline{x} = \frac{N-1}{N}(x_s + OV_s) + \frac{1}{N}\left((x_s/2) + OV_s\right) \tag{3.9}$$

$$C_s^2 = \frac{N-1}{N}\left(\frac{(x_s + OV_s)^2 - \overline{x}^2}{\overline{x}^2}\right) + \frac{1}{N}\left(\frac{\left|\frac{\overline{x}^2}{3} + \left(\frac{4x_s OV_s}{3}\right) + (OV_s)^2 - \overline{x}^2\right|}{\overline{x}^2}\right) \tag{3.10}$$

3.3 Linear Algebraic Queueing Theory (LAQT) Performance Models for SOA

The network model using LAQT built here is based on the premise that each traffic flow between nodes is assumed to be renewal and can be represented by an ME distribution. It should be noted that future work is in progress to extend the model to non-renewal traffic flows. Looking at Figure 10, we see a small network of general servers and renewal

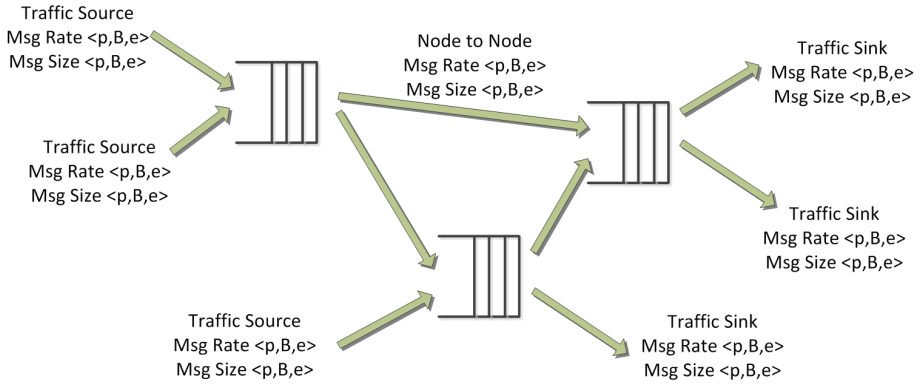


Figure 10: Network LAQT Performance Model.

traffic flows represented as ME distributions of interarrival times and message sizes between the nodes. It is assumed to be a feed forward network for each flow but not necessarily for the aggregate flows.

Figure 11 shows how an LAQT node is constructed. Traffic flows enter the node. The interarrival processes are superposed. The message sizes are mixed probabilistically to create the output message distributions. In addition, the aggregate message size distribution is used in conjunction with the rate of the FIFO queue to create the service distribution for the queue. The superposed traffic flow is serviced by the FIFO queue. The departure traffic flow is processed using a moment matching algorithm to reduce the state space of the representation of the traffic flow. Finally, the traffic flow interarrival distribution is probabilistically split or duplicated according to routing tables for transmission to the next node.

The building blocks of LAQT models are the Matrix Exponential Distribution (ME). In Lipsky's work [91], we see the definition of the ME distribution as given by its

probability distribution (PDF) as (3.11) and its CDF as (3.12).

$$f(x) = \mathbf{p}exp\left(-\mathbf{B}x\right)\mathbf{Be}' \tag{3.11}$$

$$F(x) = 1 - \mathbf{p}exp\left(-\mathbf{B}x\right)\mathbf{e}' \tag{3.12}$$

The starting vector is known as $\mathbf{p}$. The progress rate matrix is $\mathbf{B}$, and $\mathbf{e}'$ is considered a summing operator usually consisting of all ones, but not necessarily so as long as it can be use to construct a valid probability distribution. The *n*th moment in an ME distribution is given as $E\left[X^n\right] = n!\mathbf{pV}^n\mathbf{e}'$. Where $\mathbf{V}$ is $\mathbf{B}^{-1}$ (the matrix inverse of $\mathbf{B}$). ME distributions have rational Laplace-Stieltjes transforms. Our notation to describe an ME distribution is $\langle \mathbf{p}, \mathbf{B}, \mathbf{e} \rangle$. $\mathbf{L}$ is the event generator matrix and for a renewal process, $\mathbf{L} = \mathbf{Be}'\mathbf{p}$. The Kronecker space is used throughout the LAQT model. It is denoted as $\widehat{\mathbf{B}}_{\mathbf{1}} := \mathbf{B}_{\mathbf{1}} \otimes \mathbf{I}_{\mathbf{2}}$.

3.3.1 LAQT Algorithm

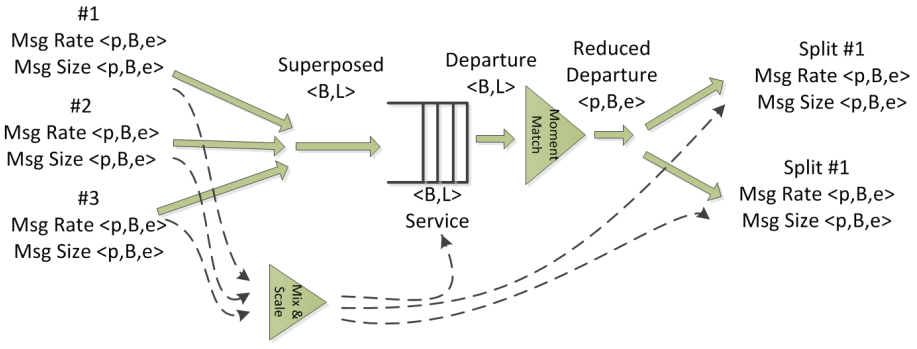


Figure 11: LAQT Node Model

The process to evaluate a network using LAQT tools is described in Algorithm 2. The pre-processing portion involves a first order analysis to determine mean arrival rates at all nodes for all products and aggregated. This is noted as λ_{pn}, where p is the product and n is the node. The steady state service process at each node is also calculated using the mean arrival rates of each product and the service process of each product at that node.

At this point node processing begins. A loop is setup that stops when the C^2 of the departure process at all nodes ceases to change during the iterations. The nodes are processed incrementally. To process a node, the first step is to determine the superposed arrival process from the input arrival processes. If a node input from an adjacent node has not been calculated yet, assume exponential distribution at the mean rate from that node. The next step is to determine the departure process followed by a moment match to reduce the state space. Finally, the departure process is split/duplicated based on the products that progress to the various next nodes.

Algorithm 2 LAQT Model

1: Compute service distribution $\langle p, B, e \rangle_s$ at all nodes n
while Departure $(C_i^2 - C_{i-1}^2) > err$ **do**
  for all n in N Nodes **do**
    2: Compute superposition process $\langle B, L \rangle_{sup}$
    3: Compute departure process $\langle B, L \rangle_{dep}$
    4: Compute reduced departure $\langle p, B, e \rangle_{mm}$
    5: Compute traffic flows out $\langle p, B, e \rangle_{n,m}$
  end for
end while
6: Compute Waiting Time Distribution at all nodes

* if a node input from an adjacent node has not been calculated yet, assume exponential distribution $< 1, \lambda_{n-1}, 1 >$

3.3.2 Step 1: Compute Service Distribution at all Nodes using ME Mixing

To represent message size distributions after a superposition operation, a probabilistic *mixing* operation takes place. The equations used to mix the message sizes for N flows are (3.13), (3.14). Since the message distribution is known, the products (and their respective arrival rates) that pass through each node are also known. We also assume feed forward message distribution.

$$\mathbf{p_{mix}} = \begin{bmatrix} \frac{\lambda_1}{\sum \lambda_i}\mathbf{p_1} & \frac{\lambda_2}{\sum \lambda_i}\mathbf{p_2} & \dots & \frac{\lambda_N}{\sum \lambda_i}\mathbf{p_N} \end{bmatrix} \tag{3.13}$$

$$\mathbf{B_{mix}} = \begin{bmatrix} \mathbf{B_1} & 0 & ... & 0 \\ 0 & \mathbf{B_2} & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & \mathbf{B_N} \end{bmatrix} \qquad \mathbf{e_{mix}} = \begin{bmatrix} \mathbf{e_1} \\ \mathbf{e_2} \\ ... \\ \mathbf{e_N} \end{bmatrix} \tag{3.14}$$

To create the service distribution used in the departure process and waiting time process at each node, the message size distribution is scaled by the FIFO queue rate at that node as shown in Figure 11

$$\mathbf{B_s} = qrate \times \mathbf{B_M} \tag{3.15}$$

3.3.3 Step 2: Compute Superposition Process

The superposition of renewal processes is described by Lipsky in [91]. We follow that approach in our work. (3.16) describes the equations to construct the superposed traffic flow where $\mathbf{Q_i} = \mathbf{e_i'p_i}$. The output of the superposition process $< B, L >$ can be sent to the departure process directly, which include any correlation information from the

superposition process.

$$\mathbf{B_{sup}} = \widehat{\mathbf{B_2}} + \widehat{\mathbf{B_2}} \qquad \mathbf{e'_{sup}} = \widehat{\mathbf{e'}}_{\mathbf{1}} + \widehat{\mathbf{e'}}_{\mathbf{2}}$$
$$\mathbf{L_{sup}} = \widehat{\mathbf{B}}_{\mathbf{1}}\widehat{\mathbf{Q}}_{\mathbf{1}} + \widehat{\mathbf{B}}_{\mathbf{2}}\widehat{\mathbf{Q}}_{\mathbf{2}} \tag{3.16}$$

3.3.4 Step 3: Compute Departure Process

In [81], Kumaran et. al. provided the approach to produce a finite departure process of an ME/ME/1 queue. This was an extension of work in [91]. This approach is used to construct the departure process in the node. We show this in (3.17) and (3.18). The $\mathbf{B_b}$ matrix is the $\mathbf{B}$ matrix from waiting time distribution, which is described in a following section. The resulting $\mathbf{B}$ matrix is finite based on the approximation in the lower right corner of the matrix [81]. The number of $\left(\widehat{\mathbf{B}}_{\mathbf{a}} + \widehat{\mathbf{B}}_{\mathbf{s}}\right)$ to include is optional and is referred to here as the departure level. After considerable simulation, departure level of 4 was determined to be adequate for this work. Choosing the correct level is still a subject of research.

$$\mathbf{B_d} = \begin{bmatrix} \mathbf{B_a} & \mathbf{L_a}\widehat{\boldsymbol{\rho}}_{\mathbf{s}} & 0 & \dots & 0 \\ 0 & \left(\widehat{\mathbf{B}}_{\mathbf{a}}+\widehat{\mathbf{B}}_{\mathbf{s}}\right) & \widehat{\mathbf{L}}_{\mathbf{a}} & 0 & 0 \\ 0 & 0 & \left(\widehat{\mathbf{B}}_{\mathbf{a}}+\widehat{\mathbf{B}}_{\mathbf{s}}\right) & \widehat{\mathbf{L}}_{\mathbf{a}} & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \left(\widehat{\mathbf{B}}_{\mathbf{a}}+\widehat{\mathbf{B}}_{\mathbf{s}}\right) & \widehat{\mathbf{L}}_{\mathbf{a}} \\ 0 & \dots & 0 & 0 & \left(\widehat{\mathbf{B}}_{\mathbf{a}}+\widehat{\mathbf{B}}_{\mathbf{s}}-\widehat{\mathbf{L}}_{\mathbf{a}}\right) \end{bmatrix} \tag{3.17}$$

$$\mathbf{L_d} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ \widehat{\mathbf{L}}_{\mathbf{s}}\widehat{\mathbf{e'}}_{\mathbf{s}} & 0 & 0 & \dots & 0 \\ 0 & \widehat{\mathbf{L}}_{\mathbf{s}} & 0 & \dots & 0 \\ 0 & \dots & \widehat{\mathbf{L}}_{\mathbf{s}} & 0 & 0 \\ 0 & \dots & 0 & \widehat{\mathbf{B}}_{\mathbf{b}}\widehat{\mathbf{e'}}_{\mathbf{b}}\widehat{\boldsymbol{\rho}}_{\mathbf{s}} & \left(\widehat{\mathbf{B}}_{\mathbf{s}}-\widehat{\mathbf{B}}_{\mathbf{b}}\right)\widehat{\mathbf{e'}}_{\mathbf{b}}\widehat{\boldsymbol{\rho}}_{\mathbf{s}} \end{bmatrix} \tag{3.18}$$

To convert the resulting MEP process described by $< \mathbf{B}, \mathbf{L} >$, to an ME distribution, we compute the steady state distribution which gives new $\mathbf{p}$ and $\mathbf{e}$ matrices [91]. From (3.17), it is clear that the size of the matrix is proportional to the size of the Kronecker space of the $\mathbf{B}$ matrix multiplied by the departure level.

3.3.5 Step 4: Compute Reduced Departure using Moment Matching

As the size of the network grows, the state space of the traffic flows grows. To reduce that state space to a manageable level, the node employs a moment matching algorithm that uses the moments of the incoming traffic flow to generate a new ME distribution for the arrival process. The moment matching algorithm is based on work by Van de Liefvoort in [151]. The equations generated by the moment matching algorithm for a three moment match are (3.19), (3.20), (3.21) where r_n is a reduced moment of the random process X, $(r_n = E[X^n]/n!)$.

$$\mathbf{p_{mm}} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{3.19}$$

$$\mathbf{B_{mm}^{-1}} = \begin{bmatrix} r_1 & r_1 \\ -\frac{r_1^2 - r_2}{r_1} & -\frac{r_1^3 - 2r_1r_2 + r_3}{r_1^2 - r_2} \end{bmatrix} \tag{3.20}$$

$$\mathbf{e_{mm}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.21}$$

The use of this algorithm for a second degree distribution is described in [110]. To use a second degree distribution, there are a few important bounds that are need to be mentioned. First, the second moment must be such that $c^2 \geq 1/2$. Second, if $1/2 \leq c^2 \leq 1$, (3.22) provide bounds for the third moment. If $c^2 \geq 1$, (3.23) applies.

$$\frac{1}{2}\left(3c^2 - 1 + (1 - c^2)\sqrt{2 - 2c^2}\right) r_1^3 \leq r_3 \leq c^2(r_1^3) \tag{3.22}$$

$$r_3 \geq 1/4(c^2+1)^2 r_1^3 \tag{3.23}$$

It is possible with more deterministic distributions, that these bounds are violated. The choices at that point are to either increase the degree of the moment match that is used or make approximations based on the situation. In this work, the following approximations are used.

- if $c^2 \leq 1/2$, an Erlang distribution is constructed with size such that the SCV of the distribution is smaller than the target SCV. The rates to proceed to the next state versus exit in the B matrix are then adjusted to match the target SCV.
- if r_3 violates (3.22), r_3 is chosen in the middle of the bounds.
- if r_3 violates (3.23), r_3 is chosen near the bound.

In all cases, the constructed ME distributions are a close approximation to the original distribution.

3.3.6 Step 5: Compute Output Traffic Flow using Splitting

To accomplish the splitting or *thinning* of the resulting renewal processes by a probability α, we simply use the (3.24) to thin a renewal process, where $\mathbf{L} = \mathbf{B}\mathbf{e}'\mathbf{p}$.

$$\mathbf{B}_\alpha = \mathbf{B} - (\mathbf{1} - \alpha)\,\mathbf{L} \qquad \mathbf{L}_\alpha = \alpha\mathbf{L} \tag{3.24}$$

3.3.7 ME Waiting Time

To calculate the waiting time distribution across the node, we rely on the work by Van de Liefvoort [152] and Kumuran et al. [82]. The waiting time distribution of our $ME/ME/1$ queue is found using the algorithm in [82]. For brevity, this algorithm is not reviewed in its entirety here. A brief description follows.

$$\mathbf{C} = \begin{bmatrix} \mathbf{B_s} & -\mathbf{B_s}\mathbf{e}'_\mathbf{s}\mathbf{p_a} \\ \mathbf{B_a}\mathbf{e}'_\mathbf{a}\mathbf{p_s} & -\mathbf{B_a} \end{bmatrix} \tag{3.25}$$

C is known as the *coupling* matrix. The eigenvalues in the negative half-plane $(\omega_1, \omega_2, ...\omega_k)$ can be used to construct the $\mathbf{B_w}$ matrix of the waiting time distribution. The associated left eigenvectors are *stacked* to form the $\mathbf{X}$ matrix used here.

$$\mathbf{B_w} = \mathbf{X^{-1}} diag\left(\omega_1, \omega_2, ...\omega_k\right) \mathbf{X} \tag{3.26}$$

$$\mathbf{p_w} = \mathbf{p_s}\mathbf{V_s} \qquad \mathbf{e_w} = \left(\mathbf{B_s} - \mathbf{B_w}\right)\mathbf{e}'_\mathbf{s} \tag{3.27}$$

3.3.8 LAQT Complexity Considerations

As was discussed, the primary reason to use the moment matching techniques is to reduce the state space of the departure matrix or the superposed matrix. Since the departure process performs a waiting time calculation which requires eigenvalue decomposition and additionally the moment matching process requires matrix inversion, $O(p^3)$ operations are performed during departure and moment matching operations. If the model converges in m iterations, we would anticipate the complexity to be $O(2 * m * n * p^3)$, where n is the number of nodes and p is the size of the departure matrix. This emphasizes the need to control the state space size.

3.4 Pub/Sub Simulation Model

OPNET from Riverbed [118] was used to create the simulation model of the Pub/Sub message pattern as shown in Figure 13. Two standard OPNET components were modified for this model. The message source is typical OPNET packet generator with a modification to allow for the use of ME distributions both in the arrival process and in the message size process. A modification for general ME distributions was proposed by the authors in [57]. Configuration is via the OPNET attributes for the specific source. Figure 12 shows a typical hyper-exponential distribution with $< p, B, e >$ matrices as shown in (3.28). The B matrix is entered as a single line with columns and rows separated by spaces. The ME_use attribute can note exponential, hyper-exponential, erlang, or general ME distributions. This enables the RNG to select the most efficient method of generating that random variate.

$$p = \begin{bmatrix} 0.788 & 0.2113 \end{bmatrix} \quad B = \begin{bmatrix} 52.57 & 0.0 \\ 0.0 & 14.09 \end{bmatrix} \quad e = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{3.28}$$

The routing module is a typical OPNET first come first serve (FCFS) server with a modification to duplicate messages to all required output ports based on input from a static text based routing file. The format of the routing file is a, b, c. Where a is the node affected by that route, b is the product affected by that route, and c is the output port for that route. To duplicate a message to two ports requires duplicate routing entries with the same a and b values and different c values. The simulation models and associated random number generators (RNG) are described in detail respectively in Appendix A and Appendix B.

Attribute	Value
name	Source 8
src 1.Arrival B	52.57308 0 0 14.08692
src 1.Arrival M1	0.0
src 1.Arrival M2	0.0
src 1.Arrival M3	0.0
src 1.Arrival ME Size	2
src 1.Arrival ME_use	3
src 1.Arrival e	1 1
src 1.Arrival p	0.78868 0.21132
src 1.Node	8

Figure 12: Source Configuration Attributes

3.5 SOA Performance Results

The analytic and simulation models were tested against the 9 node network shown in Figure 9. It should be noted that parameters other than waiting time, as shown in this paper are captured with the analytic and simulation tools. They include the squared coefficient of variation (C^2) of the waiting time and of the arrival/departure processes at each of the nodes. Scenarios were chosen that utilized just one product and 12 products. Arrival and message size distributions were varied between exponential and more general distributions. The message size distribution at each node is converted to the service

Table 2: Scenario 1-3 configuration.

Scn.	Prod.	Arr Rate	Arr C^2	Msg Size	Msg C^2
1	12	0.0667	1.0	2000000	1.0
2	12	66.7	1.0	2000	1.0
3	12	66.7	8.0	2000	3.0

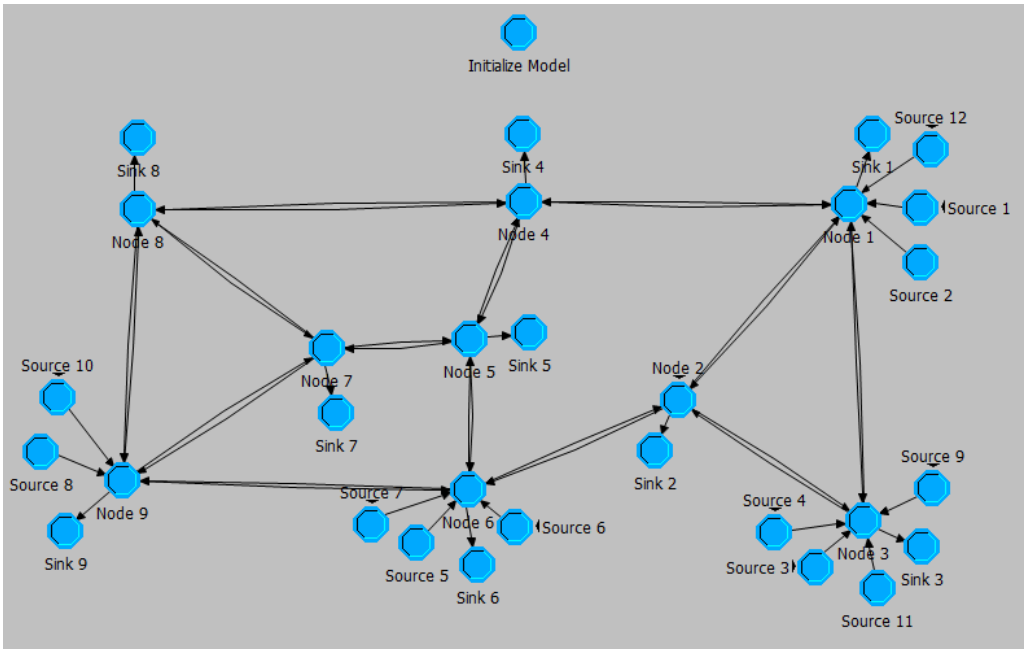


Figure 13: Opnet Modeler 9 Node Model

distribution using the queue rate. The product routing used for all scenarios is shown in Table 3. Generally, shortest path routes were chosen to the destinations. The products were selected to mimic a *real life* routing of air traffic products in an Air Traffic Control SOA Message Network. In all scenarios, the queue rates were set for 0.1 to 0.9 utilization at increments of 0.2. LAQT, QNA, and Simulation models were executed on all scenarios. Each model scenario was simulated 10 times for at least 30,000 simulation seconds each. The resulting confidence intervals are shown on the figures. With the exception of utilization levels of 0.9, the confidence intervals were small compared to the mean. The errors between simulation and analytical were generally outside the confidence intervals for higher utilization scenarios.

The purpose of Scenarios 1-3 is to evaluate the effect of cascaded servers with

Table 3: Scenarios 4-5 configuration.

		Scenario 4		Scenario 5	
		Arrival	Message	Arrival	Message
Prod.	Src-Dest(s)	Rate $C^2 = 1$	Size $C^2 = 1$	Rate $C^2 = 1$	Size $C^2 = 1$
1	1-1,2,3,4,5,6,7,9	16.67	500	16.67	500
2	1-1,2,6	0.1667	100000	0.1667	100000
3	3-1,2,3,4,5,6,7,8,9	1.333	1000	1.333	1000
4	3-1,2,3,5,6,7,9	10.0	250	10.0	250
5	6-1,2,6,8,9	0.333	250000	33.33	2500
6	6-2,6	1.333	64000	133.3	640
7	6-1,2,3,4,5,6,7,8,9	100.0	2000	100.0	2000
8	9-1,2,3,4,6,7,8,9	33.33	5000	333.33	500
9	3-2,3,6,9	0.0667	150	0.00667	1500
10	9-1,2,3,4,6,7,8,9	.0333	1000000	33.33	1000
11	3-1,3	0.1333	150	0.01333	1500
12	1-1,2,3,4,5,6,7,8,9	.0667	2000000	66.7	2000

various mean message sizes. The rate is messages/sec. The message size is in bits. In Scenario 1, we used large exponentially distributed message sizes. Figures 14(a) and 14(b) show the dramatic difference in end to end delay due to the change in message size and thus average service time. Notice the utilization is the same for both scenarios. Both scenarios had significant differences at high utilization. The mean waiting time across nodes 1 and 4 are shown in Figure 14(c) and Figure 14(d). Based on additional testing, it appears related to correlation between the service and arrival process generated by the departure process at node 1. This type of correlation is discussed extensively in [46] as well as in the conclusions of this paper.

Scenario 2 is the same as Scenario 1 with the exception that the messages are small by comparison. The arrival rate was increased to maintain the same data rate out of the producer. The waiting time decreased appropriately in all models. LAQT tended

Table 4: Scenarios 6 configuration.

		Scenario 6			
		Arrival		Message	
Prod.	Src-Dest(s)	Rate	C^2	Size	C^2
1	1-1,2,3,4,5,6,7,9	16.67	0.25	500	1.0
2	1-1,2,6	0.1667	4.0	100000	2.0
3	3-1,2,3,4,5,6,7,8,9	1.333	1.0	1333	2.0
4	3-1,2,3,5,6,7,9	10.0	1.0	250	1.0
5	6-1,2,6,8,9	0.333	0.25	250000	1.0
6	6-2,6	1.333	0.333	64000	3.0
7	6-1,2,3,4,5,6,7,8,9	100.0	1.0	2000	2.0
8	9-1,2,3,4,6,7,8,9	33.33	2.0	5000	1.0
9	3-2,3,6,9	0.0667	5.0	10.005	5.0
10	9-1,2,3,4,6,7,8,9	.0333	2.0	1000000	0.25
11	3-1,3	0.1333	12.0	150	3.0
12	1-1,2,3,4,5,6,7,8,9	.0667	12.0	2000000	3.0

to be closer to simulation especially as utilization grows. Scenario 3 is the same as Scenario 1 with the exception that the messages are small by comparison and have arrival and message size distributions that are hyper-exponential. LAQT tended to be closer to simulation especially as utilization grows. Simulated waiting time tended to be less than both models when utilization was high on nodes that were not node 1, which supports the idea of correlated arrival and departures out of node 1.

3.5.1 Multiple Product Scenarios

In Scenario 4, 5, and 6, all 12 products were chosen with sources and destinations as shown in Table 3 and Table 4. This will exercise the superposition and splitting operations. The product traffic characteristics are also shown. Three product paths out of 31 were chosen to illustrate here. They were products 1 (nodes 1-2-6-9), 6 (nodes 6-2),

and 12 (nodes 1-4-8-7). They represent small, medium, and large message sizes with different routing. Scenario 4 uses widely varying message sizes (across the products) with exponential arrival and departure distributions. Scenario 5 changes the messages sizes to similar sizes across the products. Scenario 6 uses widely varying message sizes with general arrival and departure distributions.

Looking at Figures 15(a), 15(b), and 15(c), we see reasonably good agreement with simulation at utilization less than 0.7. In Scenarios 4 and 5, as opposed to Scenario 6, LAQT was closer to simulation. The errors between simulation and analytical solutions tend to grow as utilization grows as would be expected.

As is shown with Figures 15(d), 15(e), and 15(f), we see also reasonably good agreement with simulation with the exception of Scenario 5 at high utilization. It appears that as message sizes are similar. As expected, we see low mean delay, but differences between simulation and analytical modeling at high utilization. The appearance is that correlation between arrival and service processes are possibly to blame. But, as the number of flows grows, the effect should decline. This will be examined as a component of future research.

Figures 16(a), 16(b), and 16(c) show reasonably good agreement with simulation at utilization levels less than 0.7. Again, in Scenarios 4 and 5, LAQT was closer to simulation. Scenario 6 was an anomaly to this pattern with simulation tracking with LAQT at lower utilization and QNA at higher utilization. Scenario 6 has more traffic variation than the other scenarios. Correlation caused by superposition of varying traffic types is suspected as the cause of the differences with simulation. This will be investigated

further in future research.

3.6 Summary

This work presents a new Linear Algebraic Queueing Theory (LAQT) network model that relies on a form of network decomposition with LAQT tools to model Publish/Subscribe messaging systems. Additionally, this work presents an extension to QNA to model the pub/sub messaging system. Previous decomposition approaches have generally used first and/or second moments to approximate network flows. The LAQT system proposed here is more accurate in most situations than QNA and enables the study of many additional interesting problems including correlation in networks. The one source to many destinations model used for Pub/Sub is also interesting as many network queueing models are work conserving. Both the LAQT model and QNA, using the customer creation feature have the ability to model one-to-many type flows. A simulation model was also created that was capable of copying messages to multiple destinations at any point in the network.

Some of the interesting observations that were made during this work include the discrepancies between simulation and the analytic models after a flow has progressed through cascaded servers. Dramatic differences in waiting time occurred during periods when widely varying message sizes and message size distributions were used by different products. This is undoubtedly related to the increase in the variation of the superposed process (when those products were superposed). We expect that the simulation model is *accurate* and that differences between analytical and simulation are assumed to be

approximations made by the analytical models, like the renewal approximation. Future research is planned that will confirm these hypotheses.

In addition to delay and waiting time, the LAQT model is capable of tracking other attributes like coefficient of variation of the departure process and correlation caused by superposition. We plan to use these features and develop new capabilities to address issues like correlation between arrival and service processes caused by tandem or cascaded servers.

One of the challenges to use LAQT models is the explosion in state space that occurs during the departure process and to a lesser extent during the superposition process. Using moment matching to mitigate this problem does accomplish the goal, however it comes at a cost of a loss of information including potential correlation information in the arrival flow. Another potential prospect of future research includes incorporating moment matching algorithms that includes correlation information.

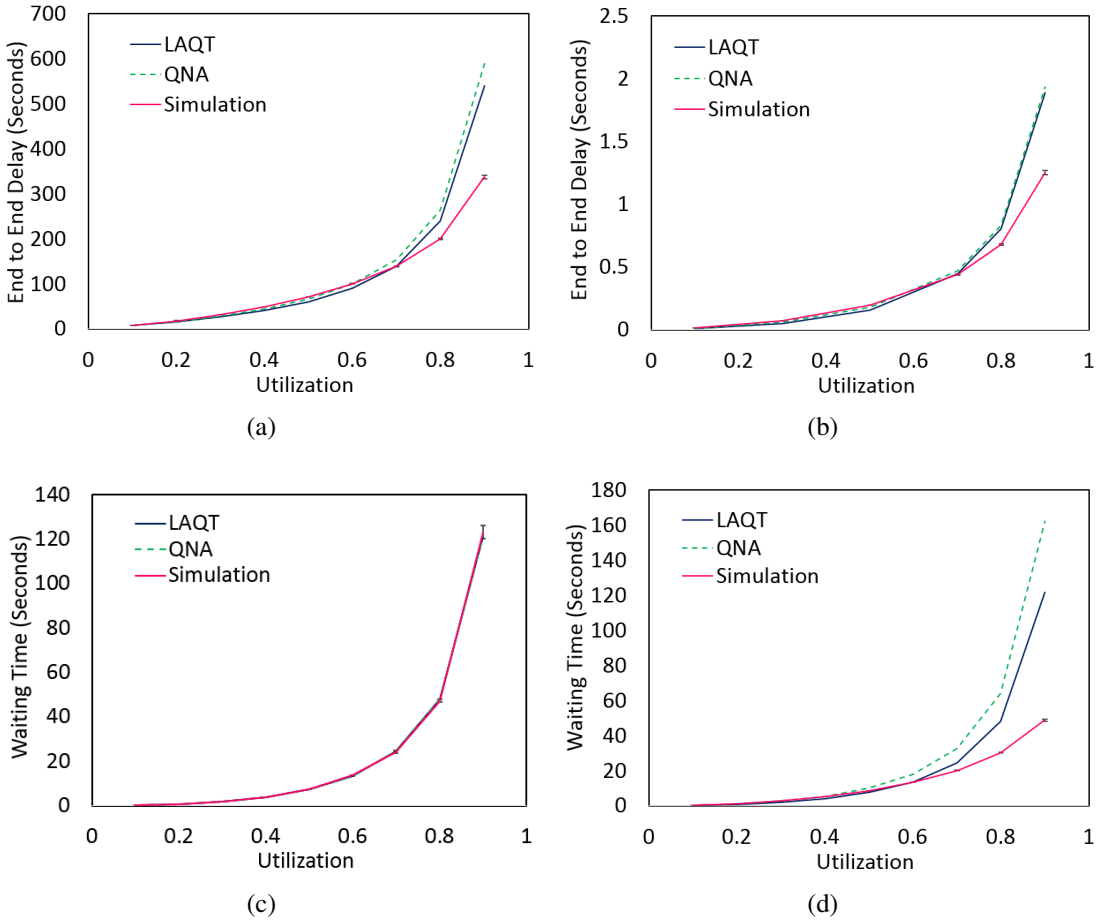


Figure 14: Single product testing (95% confidence interval for simulation at U = 0.1, 0.9). (a) Scenario 1 - single product - end to end delay (0.01, 4.3). (b) Scenario 3 - single product - end to end delay (4.36e-05, 0.017). (c) Scenario 1 - waiting time at node 1 (0.0008, 3.0). (d) Scenario 1 - waiting time at node 4 (0.0017,0.617).

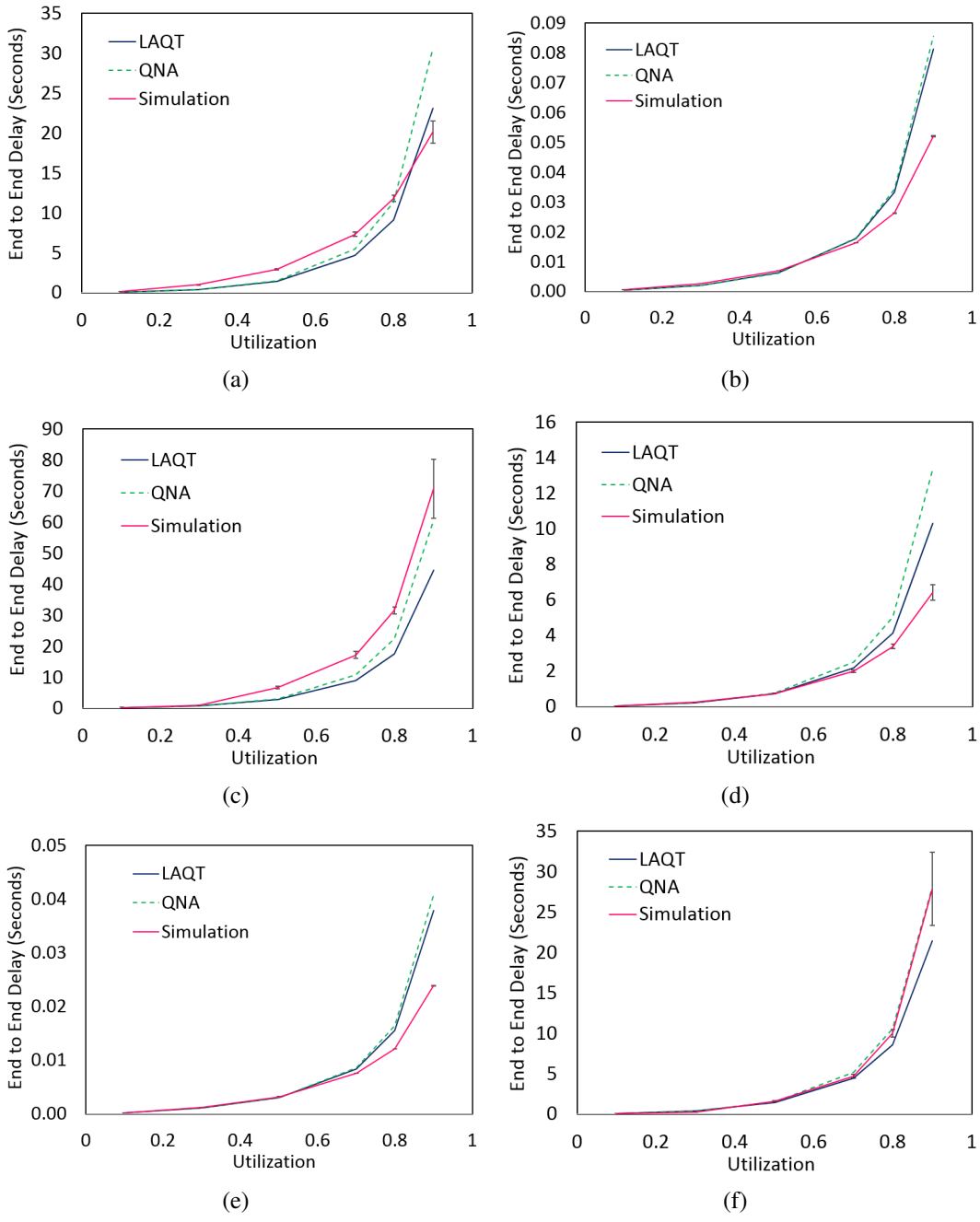


Figure 15: Multiple product testing (95% confidence interval for simulation at U = 0.1, 0.9). (a) Scn. 4 - prod. 1 - end to end delay (0.005, 1.40). (b) Scn. 5 - prod. 1 - end to end delay (6.0e-07, 1.9e-04). (c) Scn. 6 - prod. 1 - end to end delay (0.017, 9.44). (d) Scn. 4 - prod. 6 - end to end delay (5.5e-04, 0.44). (e) Scn. 5 - prod. 6 - end to end delay (1.4e-07, 6.7e-05). (f) Scn. 6 - prod. 6 - end to end delay (0.002, 4.53).

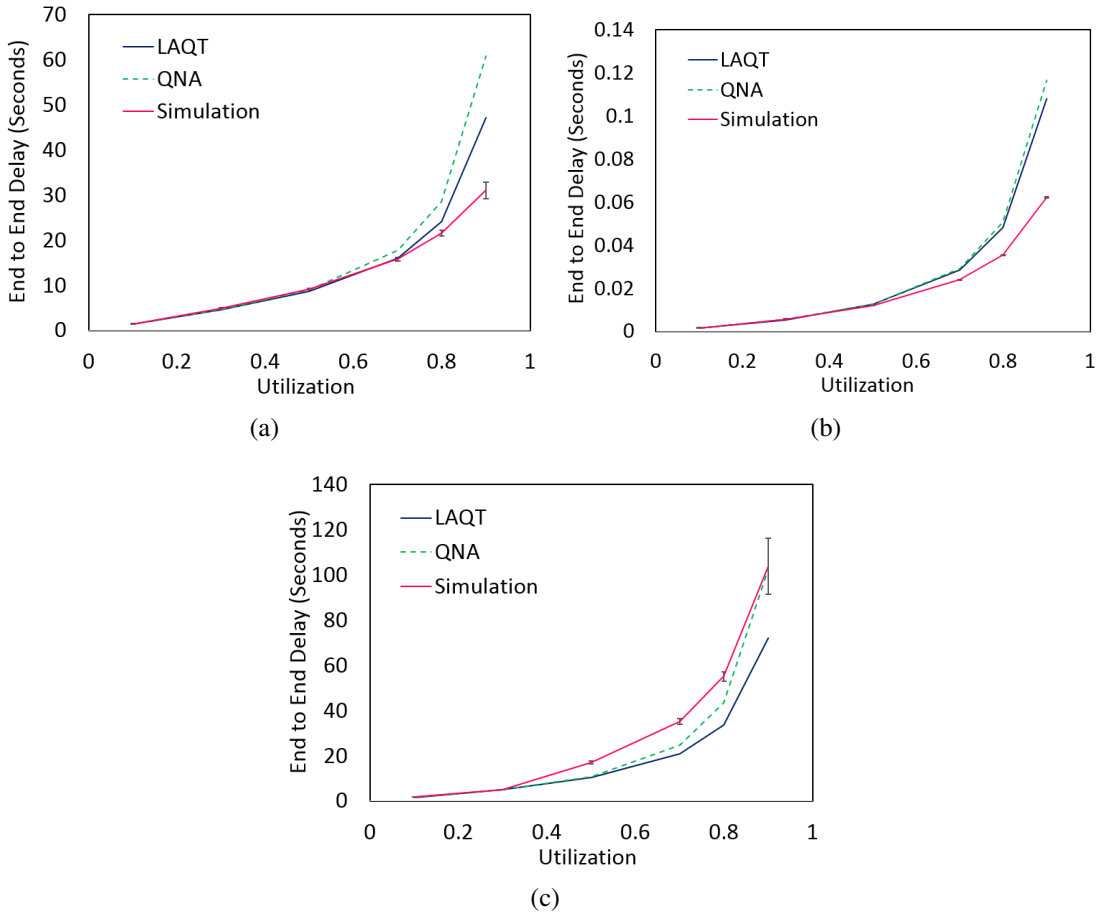


Figure 16: Multiple product testing (95% confidence interval for simulation at U = 0.1, 0.9). (a) Scenario 4 - product 12 - end to end delay (0.029, 1.8). (b) Scenario 5 - product 12 - end to end delay (7.5e-07, 1.1e-04). (c) Scenario 6 - product 12 - end to end delay (0.049, 12.52).

CHAPTER 4

EVALUATION OF GEOGRAPHICALLY CORRELATED FAILURES IN MULTI-LAYER NETWORKS

Once an effective method of mission critical network evaluation is selected, techniques are needed that can use that evaluation to determine what vulnerabilities exist in a network. This chapter looks at methods to determine if geographically correlated vulnerabilities exist and where they are located given a particular evaluation method. Work related to geographic vulnerabilities in networks frequently focuses on one aspect of the problem like topology considerations or graph theoretic network measures. The users (especially mission critical users) and the impact that a geographic event will have on those users is rarely considered.

In this chapter, we take a look at the evaluation process as it relates to geographic failures in mission critical networks. We offer the following as contributions:

- A novel impact driven, probability based resilience evaluation tool referred to as the Network Impact Resilience (NIR) metric [63].
- A self-pruning, flexible state based method, referred to as the Self-Pruning Network State Generation (SP-NSG) algorithm, to analyze multilayer networks for geographic vulnerabilities based on user requirements [63].

The NIR uses ideas borrowed from performability to combine network impact with state probability to calculate a new metric called Network Impact Resilience (NIR).

The idea is that the highest impact to the mission of a network should drive its resilience metric.

State based analysis offers tremendous flexibility in the types of network testing that are both possible and demonstrated in this work. Many other methods tend to be tied to a particular network measure. The Self-Pruning Network State Generation algorithm (SP-NSG) uses the idea of *pruning* the state space during the execution of the algorithm to reduce the states that need to be investigated, which drastically reduces computation time. One of the benefits is that it allows an effective method to identify geographic vulnerabilities in networks. Even with intelligent state space pruning methods, the analysis can become intractable for large networks. To maintain tractability in large networks, we also present a K-means clustering method. It relies on the idea of reducing the number of nodes for purposes of the state based analysis using clustering. In addition to improving tractability, we propose a method to analyze multilayer networks that simultaneously maps geographic failures onto multiple layers and analyzes the performance.

We demonstrate the accuracy of the SP-NSG with K-means clustering using simulation and observe our approach to be accurate more than 99% of the time. Additionally, the performance of the SP-NSG algorithm is reported. With proper constraints applied to the analysis, we were able to complete state based analysis techniques on very large multilayer networks that were previously intractable.

4.1 Related Work

Network resilience analysis is a varied and well studied research area. This work is a cross section of several research areas including geographically correlated failures, multiple layer networks, and network resilience metrics. We did not find existing research that generates a comprehensive set of geographic vulnerabilities in networks or resilience metrics that evaluate those vulnerabilities based on specific mission critical user requirements in multilayer networks.

For comprehensive surveys on the subject of network survivability to disasters please refer to the work by Sterbenz *et al.* [147] and Habib *et al.* [67].

4.1.1 Geographic Vulnerabilities in Networks

Much of the work related to finding geographic vulnerabilities in networks focuses on reachability of network components (or connectivity) or degradation of network capacity during a given geographically correlated failure [14, 89, 114, 128, 137]. There is also a significant amount of work that focuses on graph theoretic approaches that is discussed in Section 4.1.3.

Banerjee, Shirazipourazad, and Sen [14] presented an algorithm that uses intersections of regions around nodes and links to create a complete set of regions that can be used to analyze region based failures in networks. Li, Wang, and Jiang [89] expand on the work in [14] by considering multiple regions and link capacity.

Rahnamay-Naeini *et al.* [128] use geographically correlated stochastic models to generate link failures known as the Strauss point process. Then they compute the average

two-terminal connectivity and all-terminal connectivity using Monte Carlo simulation. Sterbenz *et al.* pioneered the concept of geographic network challenges using simulation as described in [145]. Our work is not stochastic in nature; rather, it produces all possible failure modes. Naumayer and Modiano [114] present work to calculate probabilities of geographically correlated failures in the presence of randomly placed line segments that can intersect network links. Recently, Saito [137] used probabilistic geometric models to determine geographic vulnerabilities in networks. In [1, 2], Agarwal *et al*. create hippodromes around network components based on an attack radius. They were able to use geographically correlated events to estimate the probability of failure of these components using average two-terminal reliability and capacity degradations.

4.1.2 Multilayer Network Survivability

A user requirement can be protected using end-to-end protection or link (or node) level protection. These protection techniques can implemented at the optical layer (wave protection or SRLGs), hybrid data link/network layer (Multiprotocol Label Switching (MPLS) or Generalized MPLS), network layer (IP rerouting), or higher layers using techniques like virtual network embedding (VNE) or virtual topologies. Coordination between layer protection is typically based on technology reroute times with tradeoffs, as described in [51] and [33].

Survivability of multilayer networks has been addressed before [99, 102, 121, 145, 153], but not significantly from the perspective of mission critical networks or for large geographically correlated events except by means of simulation.

State based approaches include Oikonomou, Sinha, and Doverspike who present a multilayer model in [116] that relies on the most probable states first to minimize the number of states analyzed. In our work, we utilize and extend the algorithms in [56] that use state space pruning instead of most probable states. In [88], Lee and Modiano propose a method of choosing diverse routing in multilayer networks using the Max-Flow Min-Cut theorem. In [120], Pacharintanaku and Tipper develop a multilayer mapping approach that uses lower layer mappings to provide diverse paths for upper layer connections and use mathematical programming to assign capacity.

Early work in disjoint path selection with Shared Risk Link Groups (SRLG) was completed by Oki *et al.* [117]. The concept of a probabilistic SRLG (P-SRLG) in order to find probabilities of dual path failure in the presence of correlated link failures was proposed by Lee and Modiano in [87]. In [42], Esmaelili *et al.* created multidomain Routing and Wavelength Assignment (RWA) solutions to improve lightpath reliability during correlated failures using drivers like risk minimization or traffic engineering. In [40], this was generalized to include protect paths.

In [158], Yu *et al.* proposed a Mixed-Integer Linear Program (MILP) and heuristics to map virtual infrastructure (VI) onto a substrate network so that the VI could survive certain regional failures. Also extending Virtual Network Embedding (VNE) concepts, Rahman and Boutaba [127] added survivability in order to maintain virtual networks during link failures.

In [145], Sterbenz et al. discusses an extensive frameworks in detail that describe mapping techniques from the physical layer to the application layer for the purposes of

resilience modelling. They then use these techniques to generate NS-3 code that allows the simulation of a wide variety of attacks including geographically correlated attacks.

4.1.3 Network Resilience Metrics

Network resilience metrics have different meanings based on the context. When referencing applications like web services, metrics like *performability* are considered [101, 105]. When referencing topologies and network structure, it is more likely to see *graph theoretical* or *reachability* metrics.

As was proposed by John Meyer in 1980 [109] and is described in [101] and [105], *performability* is the probability that a network will perform at a given network measure. Since the NIR has a basis in performability, it is described in detail in Section 4.2. The difference between the NIR and standard performability is that highly probable events tend to dominate performability and thus the design of the network. With mission critical networks, we are primarily interested in survivability and resilience rather than performability alone. To address this concern, we propose the concept of network impact versus network measure.

Much of the work in network reliability using state enumeration techniques was by Colbourn and Ball in [37] and [13]. This work was extended by Li and Sylvester [90] to include most probable state algorithms to calculate bounds for network reliability. Jarvis and Shier [76], as well as Dotson and Gobien [41] presented methods based on the most probable states. Gomes, Craveirinha, and Martins [65] extended these ideas to multimode network models. In 1979, Dotson and Gobien [41] created methods to

calculate 2-terminal reliability based on De Morgan's law. The method works by using failure modes to reduce the state space. Our methods utilize and extend these ideas by using failed and unfeasible network states to reduce the state space.

Graph theoretic metrics including node degree, algebraic connectivity, betweeness, clustering coefficient, and more recently weighted spectral distribution can be used to evaluate geographic vulnerabilities in networks. Work that compares these metrics includes Bigdeli, Tizghadam, and Leon-Garcia [21]; Long, Tipper,and Gomes [92]; and Cetinkaya *et al.* in [29]. A complete description of the more recent weighted spectral distribution can be found in [43] and also in [29]. Of interest for geographic vulnerabilities is the clustering coefficient metric that focuses on the connectivity of a node's neighbors described in works by Sterbenz *et al.* [146] and Ahn *et al.* [3]. One of the challenges of most of these metrics is that in general they evaluate a topology for a specific metric and not an entire network with traffic demands of varying design and importance. The ability to evaluate weighted graphs improves the situation. By itself, this information is not sufficient to help find vulnerable regions in networks considering different demand structures without a brute force type of analysis. In [159], Zhang and Sterbenz determined sets of critical nodes by computing pairwise availability as specified in [133] in weighted graphs with node sets removed that were chosen using a combination of key nodes based on centrality metrics that included degree, betweenness, and closeness.

In [52], Garbin and Shortle articulated the concept of a resilience curve, where a network performance measure like network bandwidth is contrasted to a failed percentage of network assets. The intuitive effect is useful as the concavity of the curve is indicative

of the resilience. The resilience measure can then be calculated by summing the area under the curve or simply a level of damage necessary to reduce the network measure below a given threshold. A similar approach is used by Manzano *et al.* [94]. They analyzed a network measure (referred to as a QoS parameter) versus the number of failed components. They were able to compare their approach against many graph metrics and network resilience metrics. In [106], Menth *et al.* measured resilience by calculating a complementary cumulative distribution function (CCDF) and network availability based on the network load and a set of relevant failure modes.

4.2 Network Impact Resilience (NIR)

The inspiration for the NIR is rooted in performability, which is one of the metrics used to evaluate mission critical networks. Performability uses the probability of network events and the network performance associated with that event to calculate an average network performance [101]. If N is the number of network elements, S_i is the system state, $X[S_i]$ is the network measure at state S_i, and $P[S_i]$ is the probability of network state S_i, then performability is given by

$$\mathcal{P} = \sum_{i=1}^{2^N} P\left[S_i\right] X\left[S_i\right] \tag{4.1}$$

Calculating a given network measure such as dropped connections or available bandwidth for network states is well understood; these measures are used to calculate several resilience metrics and performability metrics. However, relating network measure to the *mission* of the network is considerably more challenging. To effectively do this, two pieces of information are required.

1. The minimum network measure required to support the mission of the network ($X_{\min}$)

2. An understanding of the relationship between the network mission and network measure

How to determine these factors is complex and requires knowledge of the network mission. Modeling and simulation of catastrophic events can be used to determine this information. To define *network impact*, we start with a review of the terms *network state* and *network measure*.

DEFINITION 4.1. *Network State. The network state S_i is defined as $[s_1 s_2 \cdots s_N]$ where s_i is '0' if the network component i is up and s_i is '1' if the network component i is down.*

Clearly, this creates 2^N unique network states. Each network state may also have a unique effect on the performance of the network. Typically, the performance of a network with a particular performance indicator is called the *network measure*.

DEFINITION 4.2. *Network Measure. The network measure, $X[S_i]$, is defined as the performance of a network using a given metric at network state S_i.*

From (4.1), network measure $X[S_i]$ is related to the ability of the network to meet the demands placed on the network. Typically $X[S_i]$ ranges from 0 (i.e., 0% of demands met) to 1 (i.e., 100% of demands met). Returning to the question, *is the network able to meet its mission?*, we consider the following:

DEFINITION 4.3. *Minimum Network Measure . The minimum network measure, $X_{\min}$, is defined as the Network Measure below which the network can no longer serve its mission.*

DEFINITION 4.4. *Network Impact. The network impact, $Y[S_i]$, is defined as an indication of the networks' inability to perform its mission in a given network state S_i.*

Using this definition, $Y[S_i]$ ranges from 0 to 1. When $Y[S_i]$ is 0, the network is meeting its mission fully and there is no negative impact to the mission. When $Y[S_i]$ is 1, which by definition means $X[S_i] = X_{\min}$, the impact in state S_i has caused the complete failure of the network to meet the mission. $X_{\min}$ is strongly application dependent. This work uses $X_{\min}$ values between 30% and 50%.

Consider the situation where the network is under-provisioned. In ideal conditions, the mission of the network may still be met. But, a failure in the network may decrease the available bandwidth to the point that severe congestion occurs and the mission cannot be met due to increased latency or dropped traffic. This situation would not be considered *resilient*. In the opposite situation, the network may be *over-provisioned* to the point that many failures may occur and available bandwidth can decrease significantly before an impact to the mission is noted, providing considerably more resilience. An *impact curve* can be used to visualize the relationship between the network measure and the mission of the network:

DEFINITION 4.5. *Impact Curve. The impact curve is the function created by $Y[S_i]$ when it is sorted from minimum impact to maximum impact over the possible state space S_i.*

To provide a possible mathematical interpretation for $Y[S_i]$, we begin by incorporating the minimum network measure, $X_f[S_i]$:

$$X_f[S_i] = \max\{X[S_i], X_{\min}\} \tag{4.2}$$

It is then straightforward to define the following normalized network impact function:

$$Y[S_i] = \frac{1 - X_f[S_i]}{1 - X_{\min}} \tag{4.3}$$

Fig. 17 shows an example where $X[S_i]$ is a simple decreasing linear function over all network states. $Y[S_i]$ increases linearly from 0 impact to 1 impact at the point where $X_{\min}$ is reached.

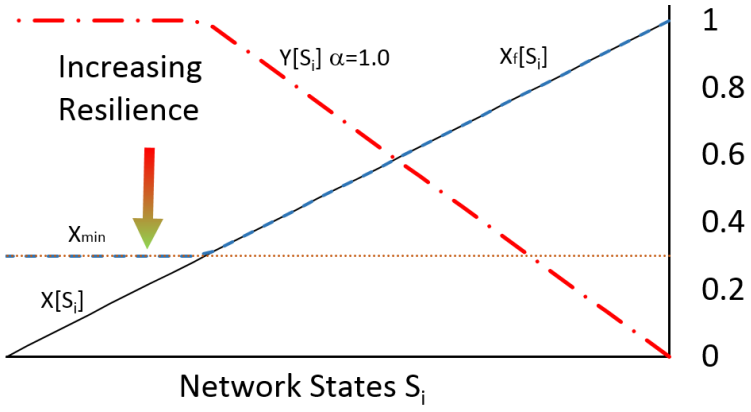


Figure 17: Network Measure and Impact ($\alpha = 1$) versus Network States S_i sorted by Decreasing Impact

To provide the ability to fit the relationship between the network mission and network measure, we add an impact parameter, α:

$$Y[S_i] = \left(\frac{1 - X_f[S_i]}{1 - X_{\min}}\right)^{\alpha} \tag{4.4}$$

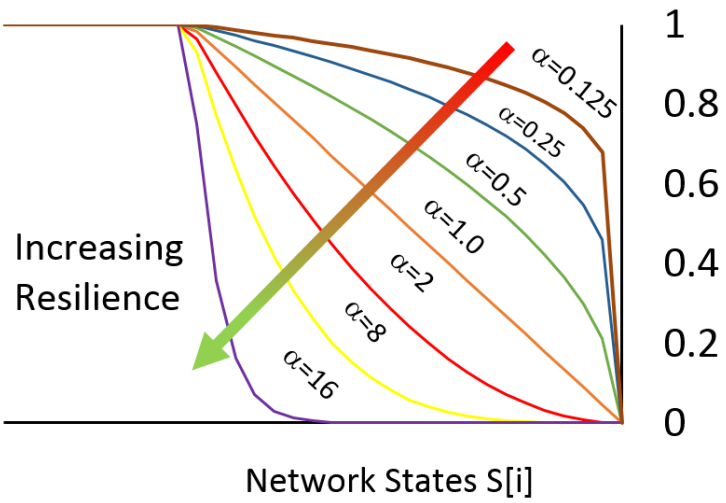


Figure 18: Impact versus Network States S_i sorted by Decreasing Impact

Note that since the minimum impact is 0 and maximum impact is 1, the impact parameter does not affect the impact at the endpoints. It will also be demonstrated that as $X_{\min}$ is decreased, the ability to improve network resilience increases.

Fig. 18 shows the effect that the impact parameter α has on the impact curves. What is useful to note here is that as α increases, the impact of the network is reduced for a larger proportion of the network state space. Therefore, it is obvious that as α increases, the resilience of the network increases. The implication here is that a network that can withstand multiple service failures prior to the inability to perform its mission would have a high α value.

Finally, the *Network Impact Resilience* is defined as:

$$\mathcal{I} = \sum_{i=1}^{2^N} P\left[S_i\right] Y\left[S_i\right] \tag{4.5}$$

Like performability (4.1), the NIR (4.5) uses state probability to determine average performance. By using impact instead of network measure, the NIR is better at predicting

the occurrence of high impact events. When $\mathcal{I} = 0$, there is no impact over the state space and would indicate that a network is impervious to failure. When $\mathcal{I} = 1.0$, it would indicate that a network cannot perform its mission. The usage of the NIR would be to compare network architectures that serve the same demands and mission. If the NIR is calculated for two networks with the same demand and mission, the lower NIR would indicate the more resilient network. This will be demonstrated in the results section.

One of the challenges with the NIR, as well as the performability, is calculating the network measure or the network impact for enough samples to accurately represent the probability of that impact or measure. Being able to represent the probability curve with a small number of samples is necessary to calculate the NIR in a tractable manner. We use $3 - 6$ samples in the most sensitive area of the probability curve and then use a piecewise approximation to construct the probabilities needed for the NIR.

4.3 Self-Pruning Network State Generation

The self-pruning network state generation (SP-NSG) algorithm was proposed in [53, 56]. It borrows on concepts from an algorithm developed by Dotson [41] to generate states as the algorithm progresses. We first present an overview of SP-NSG for single-layer networks before discussing it in the context of multilayer networks in Section 4.4. It creates a lexicographic ordering of states based on De Morgan's law where '0' denotes a working or *up* network component and '1' denotes a *down* network component:

$$\text{if } P = \{1 \;\; 2 \;\; 3 \;\; \ldots\} \text{ then } \{\overline{P}\} = \{\overline{1}\} + \{1 \; \overline{2}\} + \{1 \; 2 \; \overline{3}\} + ... \tag{4.6}$$

Nodes	1	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	…	1
	2	0	0	1	0	0	0	1	0	0	0	1	1	1	0	0	0		1
	3	0	0	0	1	0	0	0	1	0	0	1	0	0	1	1	0	…	1
	4	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	1		1
	5	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	…	1
States		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	…	32

Figure 19: Lexicographically Ordered States ('1' Denotes a Failed Node)

As stated previously, we focus on nodal failures but our work could be extended to link failures as well. In section 1, geographic vulnerability, geospatial event, and threat radius are defined. To present the algorithm, we define the following additional terms:

DEFINITION 4.6. *Feasible Mode: Based on the Threat Radius (and other filters), Feasible Mode implies that this Network State is possible.*

DEFINITION 4.7. *Network Test: Test procedure $X(S_i)$ that evaluates the performance of a network at a given Network State S_i with a Success or Fail outcome based on the ability of the network to be able to perform its mission.*

For a given Network State with a failed Network Test, the *intersection* of circles with radii equal to the Threat Radius surrounding each down node forms the Geographic Vulnerability. The following assumptions are used to develop SP-NSG.

ASSUMPTION 2. *If a network state S_i with m (≥ 1) nodes, $k_1, \ldots, k_m$, down causes the network to fail to perform it's mission, i.e., $X(S_i) < \hat{T}$ (where $\hat{T}$ is the failed threshold of*

$X_{\min}$), then any S_j that contains the same nodes will result in $X(S_j) \leq X(S_i)$. Therefore, state S_j does not need to be examined since it would also cause the network to fail its mission.

ASSUMPTION 3. *If a network state S_i with $m\ (\geq 1)$ nodes, $k_1, \ldots, k_m$, is not a feasible network state, then any S_j that contains the same down nodes will also not be feasible. Therefore, state S_j does not need to be examined.*

An example of Assumption 3 is a failure state that has *down* nodes separated by a large distance if geographically correlated failures are being analyzed, which would not need to be analyzed. We use De Morgan's Law (4.6) to reduce the search space necessary to calculate reliability [41]. As we can see from Fig. 19, State 1 can be used to generate States 6 to 9 by incrementally changing the status of each node to *down* that is after the last down node in State 1. This follows De Morgan's Law (4.6) by dividing the state space into two sections. The first section is represented by the previously successful test (in this case state 1) represented by P and the section that is untested (in this case States 6-9) represented by $\overline{P}$.

We first test the state $S_0 = [0\ 0\ 0 \ldots 0]$. If it is successful (which it should be), we use De Morgan's law to add the complement of S_0 to a list to test for both feasibility that a geographic event could affect that particular set of nodes and if that state contains a previously failed state. This would include $S_1 = [1\ 0\ 0 \ldots 0]$, $S_2 = [0\ 1\ 0 \ldots 0]$, etc... The states that are feasible and that do not contain previously unsuccessful failure modes, are added to the queue to test. The next state is popped from the queue to test and the process is repeated. If the test is unsuccessful, that state is noted. In the absence of infeasible or

unsuccessful failure modes, a lexicographic ordering is produced where the failure states are ordered by the number of failed nodes in that state as shown in Fig. 19. The SP-NSG is described in Algorithm 3 and operates as shown in Fig. 20.

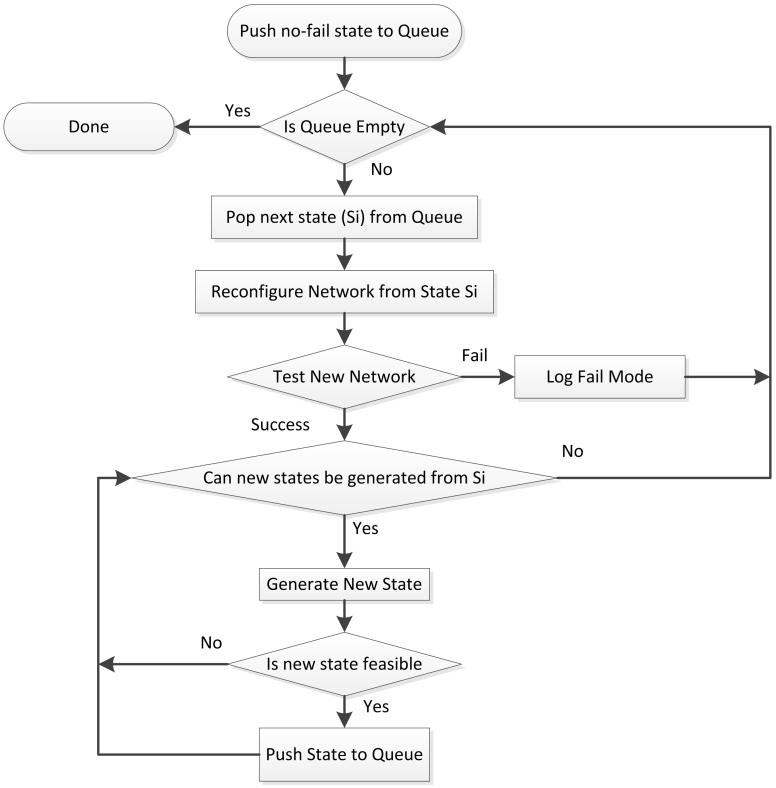


Figure 20: Self-Pruning Network State Generation (SP-NSG) Flowchart

4.3.1 Network Tests

In the step *Network Test* in Algorithm 3, we consider two different tests to determine if the network is performing its function or not under a failed state critical to its mission. These tests are 1) reachability of a network in terms of connectivity ("connectivity test"), 2) provisioning of demands for a capacitated network ("capacitated network

Algorithm 3 Self-Pruning Network State Generation (SP-NSG)

$Q \leftarrow [000\ldots0]$
while $Q \neq \{\}$ **do**
  $S_i \leftarrow Q(top)$
  $G_i \leftarrow ReconfigureNetwork(S_i)$
  if $TestNetwork(G_i) = SUCCESS$ **then**
    $N_0 \leftarrow S_i$
    while $NextLexicographicState(S_i, N_{j-1}) \neq \oslash$ **do**
      $N_j \leftarrow NextLexicographicState(S_i, N_{j-1})$
      if N_j is Feasible **then**
        $Q(bottom) \leftarrow N_j$
      end if
    end while
  else
    $FailSet \leftarrow S_i$
  end if
end while

test").

For the connectivity test, we consider the set of demands to determine if each source is connected to each destination. The metric, $M_C(S_i)$, for state S_i is defined as the ratio of successful connections over total connections. If the metric is below a certain threshold for a particular failed state, then we say that the mission critical network does not meet its requirement.

For the capacitated network test, we first provision all user demands on the network with no failures, assigning capacity to each link as needed by the demands. We consider the metric, $M_D(S_i)$, to be the ratio of the demands provisioned as successful at state S_i compared to the total traffic demands provisioned initially. If the metric is below a certain threshold for a particular failed state, then we say that the mission critical network does not meet its requirement. It should be noted that additional capacity can be added via

a redundancy factor. This is shown in the multilayer provisioning algorithm, Algorithm 5.

4.3.2 Feasible Modes

In addition to using failure modes to reduce state space, another important method to reduce state space is by using network state feasibility. If a particular network state (S_i) is not feasible, then by Assumption 3, any states that contain the same down nodes as S_i would also not be feasible. The main test for feasibility used here is *maximum geographic distance*, which is the maximum distance that any two down nodes may be located from each other. When planning for a disaster with a maximum radius of d kilometers, that disaster would not be able to simultaneously destroy two nodes that are separated by more than $2d$ kilometers. Therefore, a failure mode containing those two nodes is assumed to not be feasible under that disaster scenario. Other feasibility filters that could be used to limit state space include the maximum number of simultaneously failed nodes during a random failure analysis.

4.4 Multilayer SP-NSG Model

We first define our multilayer network model and then describe the Multilayer SP-NSG algorithm.

4.4.1 Multilayer Network Model

The network model being used includes a lower layer undirected graph denoted as $G_L(V_L, E_L)$ shown in Fig. 21. The upper layer network is modeled as an undirected graph denoted as $G_U(V_U, E_U)$. Customer demands are assumed to be given to the upper layer

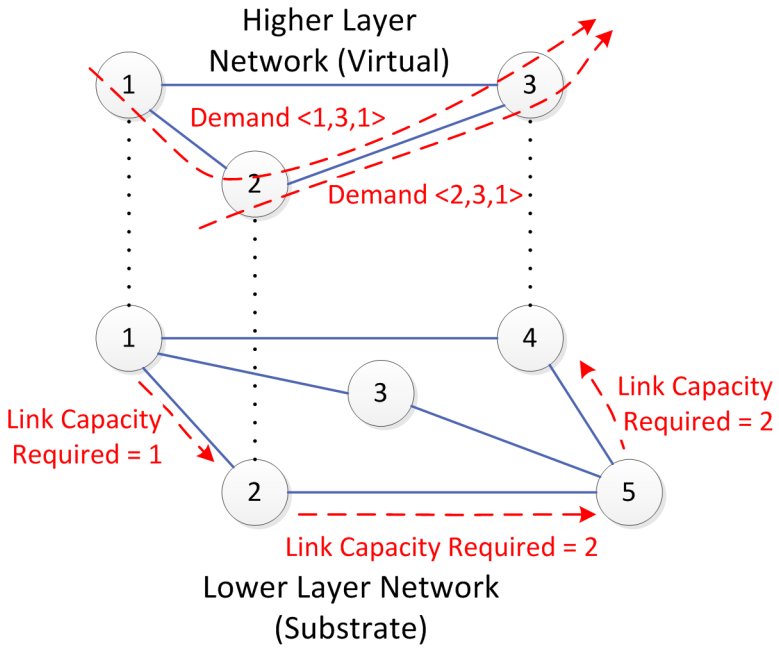


Figure 21: Multilayer Network Model

network. The i-th demand is the tuple $\langle s_U, t_U, D_U \rangle$ that indicates the demand from the upper layer source (s_U), upper layer destination (t_U), and the amount of demand (D_U). The demand $\langle s_U, t_U, D_U \rangle$ is then provisioned across G_U.

After all of the demands are provisioned, E_U now contains the link capacity required at the upper layer. E_U with the updated capacity requirements at the upper layer is passed to the lower layer network G_L for provisioning. G_L is provisioned with the upper layer link capacity requirements providing baseline. For example, in Fig. 21, two demands are sent to the upper layer network as $\langle 2, 3, 1 \rangle$ and $\langle 1, 3, 1 \rangle$. These are mapped

onto G_U and the subsequent requirements are mapped onto G_L. With shortest path routing, this leaves nodes 1-2 with a bandwidth of 1, nodes 2-5 with bandwidth of 2, and nodes 5-4 with a bandwidth of 2.

4.4.2 Multilayer SP-NSG

In Algorithm 3, the step *Reconfigure Network(S_i)* assumes that the links and nodes are based on the network state S_i. To reconfigure in the case of the multilayer network, we need to generate G_U and G_L. In order to generate new links (E_L, E_U), surviving nodes (V_L, V_U) must be generated. If we assume that the upper layer nodes will always share a location with a lower layer node, the state space for both layers can always be defined with the lower layer state space ($\mathbf{S}_i = \mathbf{S}_{L,i}$). Then a simple transformation is required to convert to the upper layer state space. This is completed with the use of an $n \times m$ transformation matrix $\mathbf{L}_{LU}$ where n is the number of lower layer nodes and m is the number of upper layer nodes. If a mapping exists between the lower and upper layer, $L_{LU}(l, u) = 1$, where l is the lower layer node number and u is the upper layer node number. All other matrix locations are 0. Converting the network state i from a lower layer to an upper one is a simple transformation as shown below.

$$\mathbf{S}_{U,i} = \mathbf{S}_{L,i}\mathbf{L}_{LU} \tag{4.7}$$

To illustrate the point, the network shown in Fig. 21 would have a transformation

matrix as shown below.

$$\mathbf{L}_{LU} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.8}$$

The algorithm for *ReconfigureNetwork(S_i)* for multilayer networks is shown in Algorithm 4.

Algorithm 4 Reconfigure Network from State S_i

for all n in V_L nodes **do**
  if $S_{i,n} = 0$ **then**
    include n in $V_{L,i}$
  end if
end for
for all k in E_L links **do**
  if $V_{L,i}$ contains end nodes for k **then**
    include k in $E_{L,i}$
  end if
end for
$\mathbf{S}_{U,i} \leftarrow \mathbf{S}_{L,i}\mathbf{L}_{LU}$
for all n in V_U nodes **do**
  if $S_{Uin} = 0$ **then**
    include n in $V_{U,i}$
  end if
end for
for all k in E_U links **do**
  if $V_{U,i}$ contains end nodes for k & path k exists in $G_{L,i}$ **then**
    include k in $E_{U,i}$
  end if
end for

4.4.3 Multilayer Network Test

The step *TestNetwork* in Algorithm 3 for single-layer networks considered either the connectivity test or the capacitated network test as discussed in Section 4.3.1.

We now describe a more sophisticated approach to perform the network test for multilayer networks that uses ideas similar to the single layer capacitated network test. In this two stage approach shown in Algorithms 5 and 6, we first provision the demands onto the upper layer network by adding capacity to the links E_U in the upper layer network. These links and their capacities would serve as demands on the lower layer network D_L. These demands are then provisioned onto the lower network creating a set of links with baseline capacities. Those capacities in the lower layer network can be augmented with a *redundancy factor* R as shown in Algorithm 5.

For the network test using a network state S_i, the upper layer links E_U are provisioned on the augmented lower layer network after it was reconfigured from S_i. The links that are provisioned successfully provide the links for the reconfigured upper layer network. The demands are provisioned on the reconfigured upper layer network. The proportion of successfully provisioned demands provide the metric as shown in Algorithm 6.

It should be noted that during the network test, links from the upper layer network may be provisioned on the lower layer network using longer paths because previously available nodes are down. This can result in more resources being allocated to the longer links which would potentially deny those resources to the shorter links. In Algorithm 6, we address this with an augmentation that searches the old list of paths generated for the links. If that path and capacity still exists, then that link is provisioned immediately. If

the path and capacity does not exist, the provisioning of that link is delayed until all of the links have been processed for the first time.

Algorithm 5 Initial Multilayer Network Provisioning

```
for all k in D_U Demands do
  if path P_{U,k} is available in G_U for k then
    for all e in E_U Links do
      if e is in P_{U,k} then
        e_cap ← e_cap + k_dem
      end if
    end for
  end if
end for
for all k in E_U do
  if path P_{L,k} is available in G_L for k then
    for all e in E_L do
      if e is in P_{L,k} then
        e_cap ← e_cap + k_cap
      end if
    end for
  end if
end for
for all e in E_L do
  e_cap ← e_cap × R Redundancy Factor
end for
```

4.4.4 Improvements to SP-NSG for Geographic Vulnerabilities in Large Networks

One of the issues with multilayer networks is that the *network test* typically has high computational costs, especially for large networks. The basic implementation of SP-NSG has a cost that is associated with the possible number of states 2^{N_r} given a maximum of N_r nodes in a particular threat radius r. However, as we know from [56], the number of states analyzed is still typically a very small fraction of the maximum number of states.

Algorithm 6 Multilayer Network Test

Reconfigure $G_{L,i} \leftarrow G_L$ for Event i
$G_{U,i} \leftarrow []$
for all k in initial E_U links **do**
  if initial $P_{L,k}$ and k_{cap} is still available in $G_{L,i}$ **then**
    for all e in E_L **do**
      if e is in $P_{L,k}$ **then**
        $e_{cap} \leftarrow e_{cap} - k_{cap}$
        $G_{U,i} \leftarrow G_{U,i} + k$
      end if
    end for
  else if path $P_{L,k}$ and k_{cap} is available in $G_{L,i}$ for k **then**
    for all e in E_L **do**
      if e is in $P_{L,k}$ **then**
        $e_{cap} \leftarrow e_{cap} - k_{cap}$
        $G_{U,i} \leftarrow G_{U,i} + k$
      end if
    end for
  end if
end for
$E_{U,i} \leftarrow G_{U,i}$
$pos \leftarrow 0, neg \leftarrow 0$
for all k in D_U Demands **do**
  if path $P_{U,k}$ and k_{dem} is available in $G_{L,i}$ for k **then**
    for all e in $E_{U,i}$ **do**
      if e is in $P_{U,k}$ **then**
        $e_{cap} \leftarrow e_{cap} - k_{cap}$
      end if
    end for
    $pos++$
  else
    $neg++$
  end if
end for
$NetMeasure \leftarrow pos/(pos + neg)$

We consider the following assumption to improve on SP-NSG for large networks:

ASSUMPTION 4. *If two nodes are located in close proximity to each other and one node fails due to a geographic event, then the other node will also fail.*

Assumption 4 raises a few questions. First, should probabilistic failures be considered? This is a question that affects the entire SP-NSG algorithm. Following the mission-critical thread for this work, when looking for geographic vulnerabilities in networks, we are less interested in what is the *likely* outcome as we are with what is the *potential* outcome if mitigating steps are not taken. To that end, Assumption 3 seems to apply. Furthermore, we are interested for the *worst case*, not the *average case*.

With Assumption 4, we propose that a *node* could represent multiple nodes that are located in close proximity to each other. The set of nodes used in the state generation algorithm can be reduced and mapping can be established to *both* the upper and lower set of nodes/links. We acknowledge that this is an approximation when the node location that is considered in the state generation is not exactly the node location in the network. However, we build on the premise that if the distance between the new node location and the mapped node is relatively small, the outcome is likely to be the same. The advantage of this approach is that the network nodes (in either layer) that are no longer included in the state generation algorithm are still used via mapping. This preserves the network connectivity and structure at those layers. To select new node locations, we use the K-means clustering algorithm similar to the algorithm described in [23] in areas of the network that are *dense* as described in Algorithm 7, which is used to create a new set of nodes V_A.

Algorithm 7 Large Network Node Reduction Algorithm using K-Means Clustering

$reduce \leftarrow yes$
$V_A \leftarrow V_L$
$L_{AL} \leftarrow [zero]_{A \times L}$
while $reduce$ **do**
  $reduce \leftarrow no$
  $maxnode \leftarrow 0$
  for all n in V_A nodes **do**
    if Number of Nodes m contained in threat radius $r_n > threshold$ **then**
      if $m > maxnode$ **then**
        $reduce \leftarrow yes$
        $maxnode \leftarrow m$
        $update \leftarrow n$
      end if
    end if
  end for
  if $reduce$ **then**
    $V_A \leftarrow$ K-Means$(V_{r_{update}})$
    $L_{AL} \leftarrow$ '1' at V_A, V_L for all V_L reduced
  end if

end while

Once V_A is known, the mapping can be generated in a manner similar to the previous section that allows the network states generated from V_A to be mapped onto V_L and V_U as follows:

$$\mathbf{S}_{L,i} = \mathbf{S}_{A,i}\mathbf{L}_{AL} \tag{4.9}$$

Note that $S_{U,i}$ is still created from (4.7).

One final optimization that was used was to store the shortest path results during the initial (all working) network test. If a new network state does not affect any of the nodes in the original shortest path, the original results would still hold and a new shortest path calculation does not need to be run. In a large network, we found that this leads to substantial improvements.

4.5 Using Clustering Models to Reduce Complexity

Without clustering, the complexity of our approach is $F(N) * N * 2^{N_r}$ where N_r is the maximum number of nodes in a threat radius, N is the total number of nodes, and $F(N)$ is the complexity of computing the network measure. When clustering is incorporated, the complexity is reduced to $F(N) * N * 2^k$, where k is the new maximum number of nodes in a threat radius after clustering. Clearly, if k can be limited to a relatively small number ($k \ll n$), then the complexity is significantly reduced.

Now consider the complexity of $F(N)$. For the multilayer case, a breadth-first shortest path calculation (BFS) is used for every link in the upper layer network across the lower layer network along with a breadth-first shortest path calculation for every user demand across the upper layer network. Therefore, $F(N) = |E_u|(|N_L| + |E_L|) + D(|N_U| +$

$|E_U|$) for a total of D demands in the upper layer network. If $|N_L| \gg |N_U|$, this simplifies to $F(N) = |E_u|(|N_L| + |E_L|)$.

4.5.1 Model Selection Considerations

The concept behind using clustering to represent portions of the system topology during the state generation algorithm is to approximate node locations with *clustered* node locations then use mapping techniques to reconstruct the topologies with the selected states. One of the challenges with K-Means Clustering is the selection of K. If K is too large, complexity reduction benefits are eliminated and the model is potentially *over-specified*. If K is too small, the model may be inaccurate.

Rissanen suggested the minimum description length (MDL) concept in [131] and is described by Principe, Euliano, and Lefebvre in [123] as a way to balance model complexity and accuracy. The basic idea is to understand the minimum set of datapoints to describe the system in question. Here we would assume that is the list of communication nodes. If we assume we need to reduce the number of nodes to represent the system, this can involve either eliminating redundancies in the data or *approximating* the system, increasing the error associated with the model. In our system, we also assume that every node in the topology is needed to fully describe the topology. therefore, any reduction in the number of nodes would be associated with an approximation.

Next we need to determine an error metric that can be used to evaluate our model. Without a complete analysis of the system (which is not always tractable), we can evaluate the system in a tractable configuration and attempt to associate it with a metric. Using

Assumption 4, we assume we can replace closely positioned nodes with a single node for purposes of generating network states. This fits a clustering model conveniently. We used a K-Means Clustering algorithm as described by Bishop in [23].

One of the challenges with the K-Means Clustering algorithm is that the selection of K is left to the user. It would seem that a larger K would be more accurate but also less tractable. In Figure 22, we see that in our system, the errors using a clustered approach follows the average Euclidean distance from each node to the selected cluster location. This would imply that the average squared distance from the clustered location (which is known) would be reasonable error metric. We will denote this as the average sum of squared error (Average SSE) and define it as shown in (4.10). L_K is the total number of clusters across the entire topology which is based on the selection of K. C_i is the set of nodes in the ith cluster (along x and y dimensions). N is the total number of nodes that were clustered and m_i is the cluster location for the ith cluster.

$$\text{Average SSE} = \frac{1}{N} \sum_{i=1}^{L_K} \sum_{n \in C_i} \left((n_x - m_{ix})^2 + (n_y - m_{iy})^2 \right) \tag{4.10}$$

Taking a look at Figure 23, we see that as expected, processing time is inversely related to Average SSE. For our purposes, we chose a variation of the *elbow* method to choose K. Basically, the location where the reduction in error associated with the increase in K begins to decrease and a steep increase in processing time was chosen. K of 4,9,16 were all tested and this is shown in Section 4.7.

An alternative method might be to use an MDL criterion like the Akaike Information Criterion (AIC) [5] or the Bayesian Information Criterion (BIC) [139] to select a K

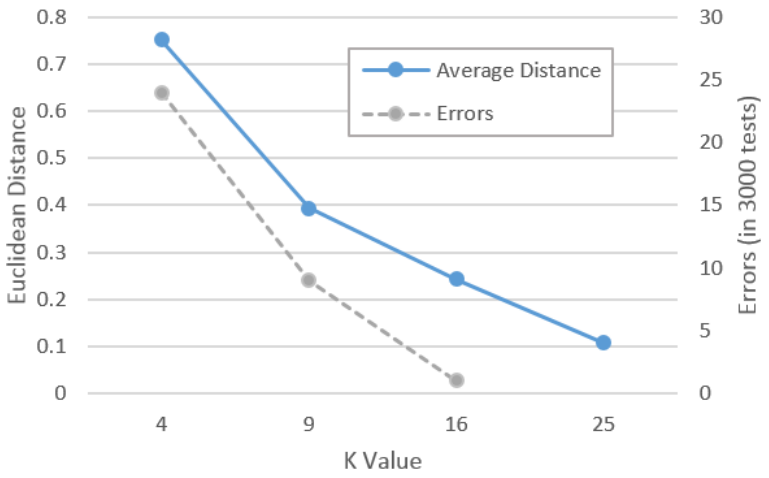


Figure 22: Comparison of Average Distance to Cluster Location and Model Errors

value that balances accuracy and model complexity. The AIC and BIC are both described in [23] and [123]. The equations used here to compute AIC and BIC are shown in (4.11) and (4.12)

$$\text{AIC}(L_K) = N\ln J(L_K) + 2L_K \tag{4.11}$$

$$\text{BIC}(L_K) = N\ln J(L_K) + \frac{L_K}{2}\ln(N) \tag{4.12}$$

Where N is the total number of nodes; $J(L_K)$ is the Average SSE; and L_K is the total number of clusters across the topology with a selection of K. Additionally, we looked at the AICc criterion as described in [25] which is shown in (4.13). The AICc criterion was created for situations where the number of free parameters is more closely related to the size of the model as is the case with our system. AIC and BIC do not really meet this criteria for our system but are included here for discussion purposes.

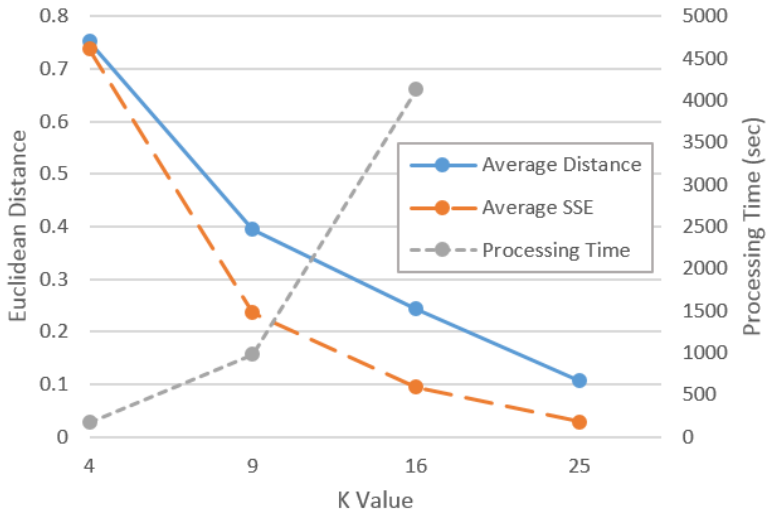


Figure 23: Comparison of Average Distance to Cluster Location, Average SSE, and Model Processing Time

$$\text{AICc} = \text{AIC} + \frac{2L_K(L_K+1)}{N - L_K - 1} \tag{4.13}$$

Figure 24 shows a comparison of the AIC, BIC, and AICc. It is clear that the AIC and BIC do not penalize the model complexity enough while the AICc penalizes it too much. This is an area that requires further study to realistically use these criterion for assessment of K.

It is possible to select clustered locations using different models completely. For example, many classifiers like Support Vector Machines (SVM) and Neural Networks (NN) [23] could be used to identify clusters of nodes that may be able to be approximated using a single node in a manner similar to how K-Means Clustering is used here. Finally, credit assignment is a method by which model components that improve the performance of the model are identified and rewarded accordingly and visa versa. This could be used

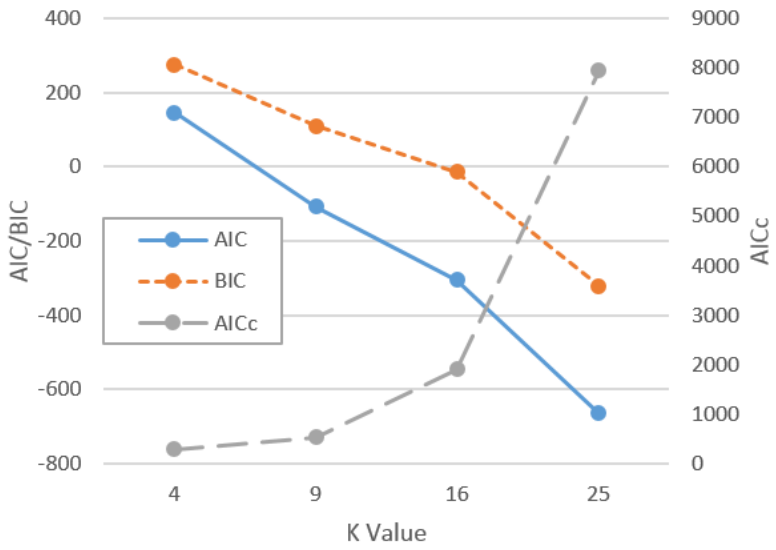


Figure 24: Comparison of AIC, BIC, and AICc

to help select K on a cluster by cluster basis [23].

Taking these concepts a step further, one could define the problem in terms of risk and loss, where the error function is considered the loss and the risk is the expected value of the loss. This would allow bounding of the problem to a certain level of loss. A bigger picture approach could even attempt to attach a monetary value to the vulnerabilities found and use that value to assess loss and therefore risk.

4.6 Evaluation Methodology

To evaluate the methods proposed here, we consider three multilayer networks representing a small, medium, and large network. The small network is a Gabriel topology with 16 nodes in the lower layer network and 6 (N1, N6, N12, N1, N5, and N14) nodes in the upper layer network. As shown in Fig. 25, we consider three different upper layer topologies: the ring topology (Topology-1), semi-mesh topology (Topology-2), and star

topology (Topology-3) with N13 as the hub.

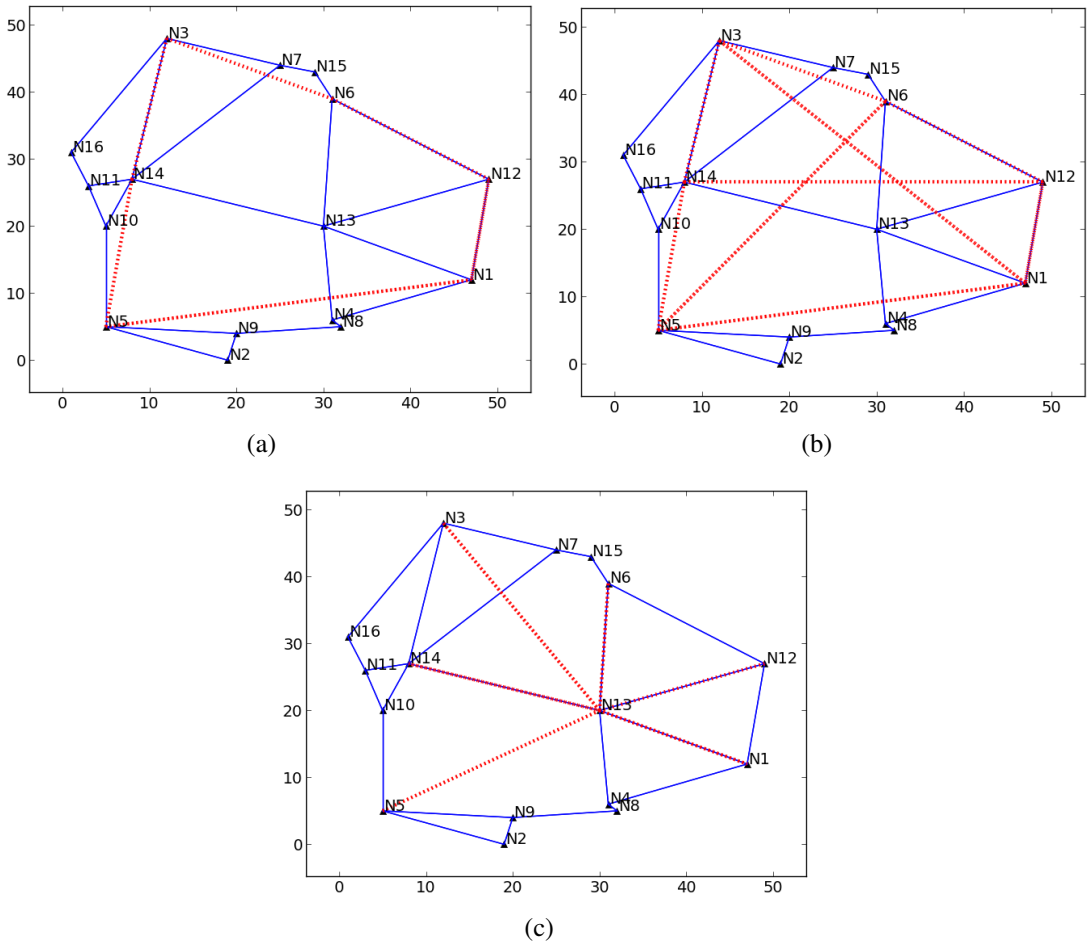


Figure 25: Gabriel Network. Lower layer network is shown with solid lines. Upper layer network is shown with dotted lines. (a) Gabriel Upper Layer Network #1 (ring) (b) Gabriel Upper Layer #2 (semi-mesh) (c) Gabriel Upper Layer #3 (star)

The medium network is the Nobel-EU topology from the Survivable Network Design Library (SNDlib 1.0) [119] with 28 nodes in the lower layer network and 10 nodes in the upper layer network. Again, three different upper layer topologies are tested: the ring topology (Topology 1), semi-mesh topology (Topology 2), and semi-star topology

(Topology 3); see Fig. 26. This multilayer network example is referred to as the EU network.

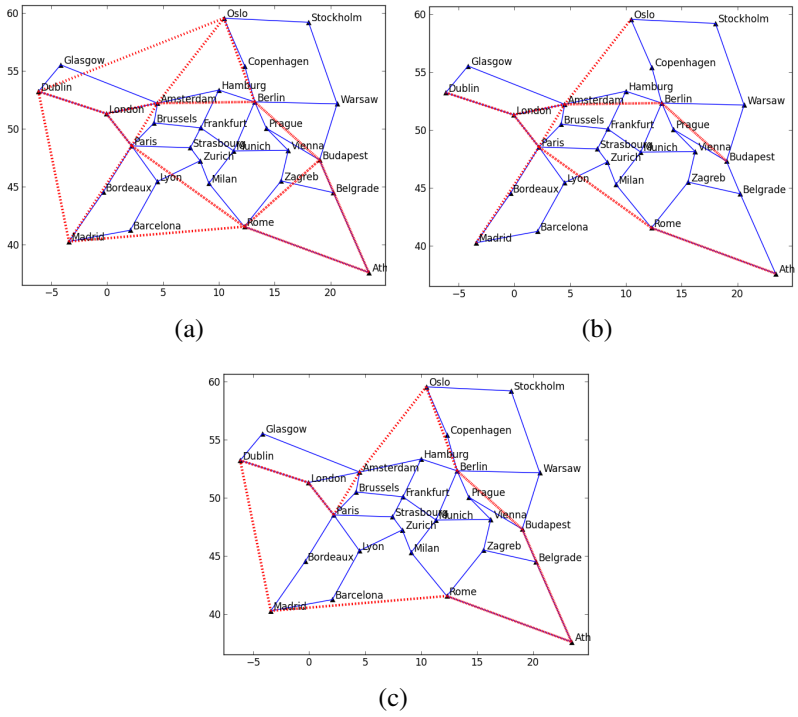
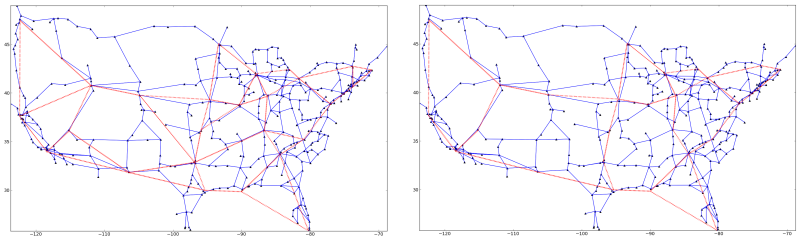


Figure 26: Nobel-EU Network. Lower layer network is shown with solid lines. Upper layer network is shown with dotted lines. (a) EU Upper Layer Network #1 (semi-mesh) (b) EU Upper Layer #2 (semi-star) (c) EU Upper Layer #3 (ring)

The large network is the AT&T Layer 1 topology generated by Sterbenz, *et al.* [145] and available at [84] . This is a large physical layer network with 383 nodes. There are two different upper layer topologies that are tested: the dense topology (Topology-1) and regular topology (Topology-2); see Fig. 27. This multilayer network example is referred to as the ATTL1 network.

All upper layer demands are simply one unit of bandwidth for each demand and the demand structure is always from all upper layer nodes to all other upper layer nodes.



(a) (b)

Figure 27: ATTL1 Network. Lower layer network is shown with solid lines. Upper layer network is shown with dotted lines. (a) ATTL1 Upper Layer Network #1 (dense) (b) ATTL1 Upper Layer #2 (regular)

These networks are analyzed using a network test that is appropriate for the topology. For example, if one node is down in the upper layer network of the Gabriel network, 10 demands become automatically out of service. If the network test required 70% demands, then one node down would cause the network to fail. So for the Gabriel networks, 40% was selected as the threshold to satisfy mission critical requirements. For the EU network, 50% was selected and for the ATTL1 network 70% was selected.

A set of 3000 random network attacks was also created to test the accuracy of the algorithms. The network test is run based on the random location of the attack and subsequent down nodes within the given Threat Radius of the attack location. They are shown with the output of the SP-NSG algorithm. There are several parameters that can be varied to conduct the analysis. These include the number of nodes allowed in a cluster (which are varied from 4 to 16), the size of the threat radius (which is topology dependent), redundancy factor (which is varied from 1 to 3), and network test parameter. Several statistics are collected with each operation including run time, states tested, states failed, vulnerable areas of the network, and ratio of vulnerable area to total area.

4.7 Results

The analysis tool that implements the SP-NSG algorithm was built and compiled using GCC C++ on an MS Windows 7 Personal Computer with an Intel I7 processor and 8 GB of RAM. The graphic output and network/configuration interfaces use an XML format. Python 3.2 was used to generate the output graphics. The random attack generator was also built and compiled using GCC C++. The XML network formats used here are very similar to those used by [119] in the Survivable Network Design Library.

For all three networks, the NSG-SP algorithm predicts the Geographic Vulnerability. The red 'o' sign denotes a random attack that did sufficient damage to cause the network measure to be less than $X_{\min}$ causing the mission to fail. The gray '+' sign denotes a random attack that caused a network measure that was greater than $X_{\min}$ and thus, the mission did not fail. The shaded area should correlate with the red marker locations.

4.7.1 Gabriel Network

From Figures 28(a) to 28(e), we illustrate a few important points. First, as the threat radius grows, the vulnerable area grows. This is intuitive since with the increase in the threat radius, more nodes are affected by a single event causing more severe impacts. Figure 28(b) shows that as the threat radius grows from 20 to 25, vulnerabilities show up between nodes N5 and N14, and between nodes N1, N12, and N13. The latter failure mode illustrates the interaction between the upper layer and lower layer since N13 is only a lower layer node and nodes N1 and N12 are present in both layer topologies.

An important point is that the effect on different upper layer topologies are not

Table 5: Results for the Gabriel Network

Upper Layer Topo.	Threat Radius	Red. Factor	Vuln. Area	States Exam.	Filter States	Tested States	Succ. States	Loss States	Comp. Time	Errors
1	10.0	1	0.000	355	289	66	66	0	0.25	0
1	12.5	1	0.017	497	381	113	111	2	1.11	0
1	15.0	1	0.103	907	623	241	235	6	1.97	0
1	15.0	2	0.103	907	623	241	235	6	3.00	0
1	15.0	3	0.103	907	623	241	235	6	3.19	0
2	12.5	1	0.421	328	234	66	56	10	5.45	0
2	15.0	1	0.617	532	328	113	101	12	7.86	0
2	10.0	1	0.236	264	210	47	41	6	3.03	0
2	10.0	2	0.209	288	234	51	49	2	2.25	0
2	10.0	3	0.086	296	238	58	57	1	1.28	0
3	10.0	1	0.259	308	246	47	45	2	2.22	0
3	12.5	1	0.386	384	281	63	61	2	2.64	0
3	15.0	1	0.512	633	403	117	115	2	5.29	0
3	15.0	2	0.512	633	403	117	115	2	5.56	0
3	15.0	3	0.512	633	403	117	115	2	3.25	0

always predictable. The Gabriel #2 topology had vulnerabilities that the Gabriel #1 topology did not. This is because when the upper layer network was more flexible, more direct routes were available between nodes and they tended to utilize the same nodes (N1 and N14) while entirely ignoring other longer routes. This was determined by analyzing the flow analysis after provisioning. This suggests that some extra steps would need to be exercised to ensure that the flexibility of the upper layer network did not cause the entire network to be more vulnerable. The star topology performed as expected with a vulnerability at the center of the star. Table 5 shows the information that was collected on the Gabriel network. Computation time (sec.) and errors are discussed later in the paper.

The NIR metric shown in Table 6 reflects the vulnerabilities shown in Table 5 and in Figure 28. The Gabriel #1 topology has the least vulnerabilities with a threat radius of

15. To compute the NIR, the minimum network measure $X_{\min}$ was swept at intervals of 10% from 40% to 90%, approximating the probability curve. It is noteworthy that as α grew to 16, the differences between the topologies became more apparent. This is because the #1 topology had very few vulnerabilities when $X(S)$ was less than 50%. Figure 31 shows the impact curve and the approximated probability distribution for Gabriel #1 and #3 topologies. With topology #3, it is apparent that the higher probability of network failure at a lower network measure drives the NIR up for that topology.

Table 6: Gabriel NIR with $X_{\min} = 0.4$ for three upper layer topologies

α	Topology 1	Topology 2	Topology 3
1	0.73	0.83	0.78
4	0.33	0.68	0.57
16	0.11	0.62	0.51

4.7.2 EU Network

For the figures associated with the EU network, we use the same notation used with the Gabriel network. In a similar manner to the Gabriel network, Figures 29(a) to 29(e) illustrate that as the threat radius grew the vulnerable area grew. Perhaps the most noteworthy point is that when the upper layer network was more dense, the network was more vulnerable. We saw this same effect with the Gabriel network. Also notable is that the loop topology was as resilient to geographic events as any other topology to geographic events. Table 7 shows the information that was collected on the EU network. Computation time and errors are discussed later in the paper.

The NIR metric shown in Table 8 reflects the vulnerabilities shown in Table 7. Note that the NIR favors different alternatives for different α values. With $\alpha = 1$, the

Table 7: Results for the EU Network

Upper Layer Topo.	Threat Radius	Red. Factor	Vuln. Area	States Exam.	Filter States	Tested States	Succ. States	Loss States	Comp. Time	Errors
1	2.5	1	0.011	866	751	111	107	4	1.34	0
1	5.0	1	0.200	4190	2439	1213	1181	32	8.02	0
1	3.75	1	0.078	1969	1521	354	343	11	3.45	0
1	3.75	2	0.078	1982	1533	356	344	12	3.56	0
1	3.75	3	0.060	2019	1569	358	348	10	3.23	0
2	2.5	1	0.114	563	467	89	83	6	1.39	0
2	5.0	1	0.353	1645	939	559	537	22	8.41	0
2	3.75	1	0.227	881	673	175	162	13	3.43	0
2	3.75	2	0.227	881	673	175	162	13	3.82	0
2	3.75	3	0.223	975	745	200	190	10	3.49	0
3	2.5	1	0.031	828	714	106	103	3	1.17	0
3	5.0	1	0.200	4045	2358	1124	1099	25	12.62	0
3	3.75	1	0.089	1805	1393	316	305	11	3.40	0
3	3.75	2	0.089	1832	1417	317	307	10	3.34	0
3	3.75	3	0.089	1832	1417	317	307	10	3.60	0

NIR indicates that the EU #2 topology is more resilient. At $\alpha = 16$, the EU #1 and #3 topologies are more resilient. The reason for this non-intuitive result is that all three upper topologies perform similarly when the concern is equal for the loss of 10% to 90% of the demands. However, when the loss of 60% of the demands is weighted more heavily ($\alpha = 16$), the loop and partial mesh topologies ($\#1$ and $\#3$) are the clear favorites. The intuition for the results seems to hold.

Table 8: NIR for the EU network for three upper layer topologies

α	Topology 1	Topology 2	Topology 3
1	0.44	0.34	0.54
4	0.14	0.18	0.15
16	0.06	0.13	0.06

4.7.3 ATTL1 Network

Figures 30(a) to 30(c) show that the area of vulnerability seems to follow a line from the northeast to the southwest U.S. During the initial multilayer provisioning process, only two flows were created across the network in the western part of the country. One path was near the vulnerabilities marked and the other path was through the middle part of the country, entirely ignoring the path on the northern part of the topology. Even with redundancy augmentation, capacity was not added as no additional initial capacity was needed.

The provisioning algorithms (used in this work) use a shortest path algorithm to find paths. It may be required to investigate other methods to generate paths in order to encourage more path diversity. Interestingly, the large network tended to follow traditional assumptions, with sparse upper layer networks faring worse than more dense upper layer networks.

Figure 32 shows the results of the clustering algorithm. The dark lines are a tree with the new clustered location at the base of the tree. The nodes that are clustered are at the leaves of the tree. When the threat radius is large (in this example it is 2.0) and the cluster factor is high (9), considerable clustering occurs.

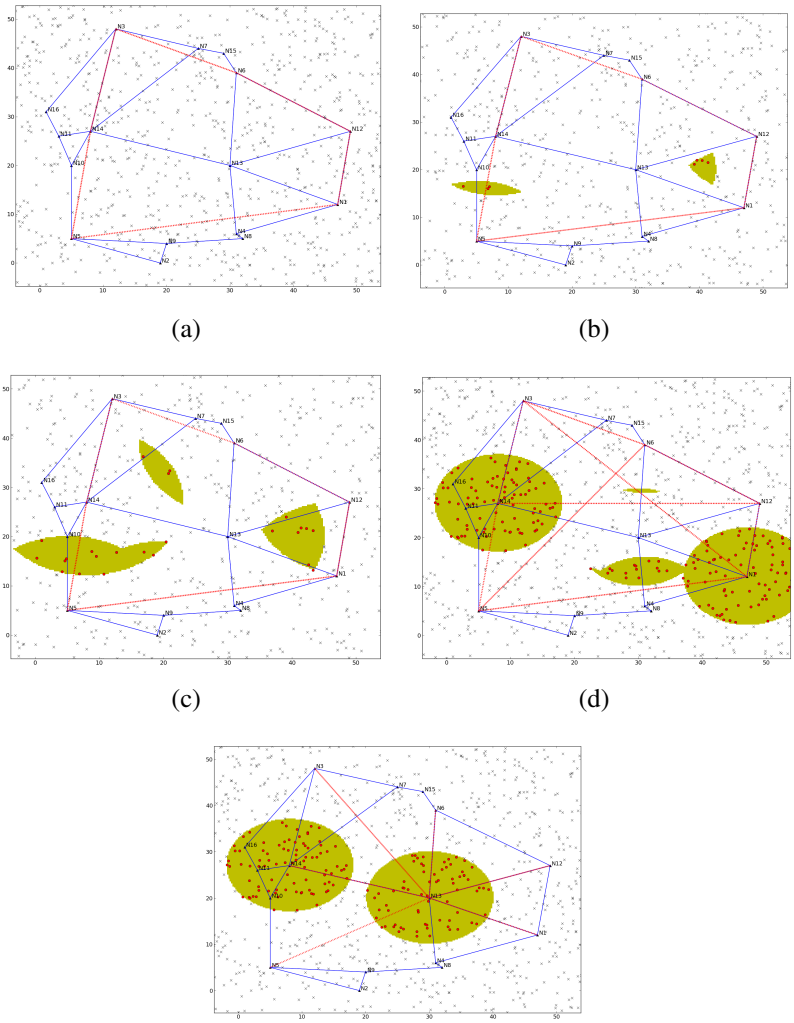
The NIR also agrees with Table 9 and Figure 30(e), which shows the most resilient topology to be topology 1.

Table 9: Results for the ATTL1 Network

Upper Layer Top.	Threat Rad.	Red. #	Clust. #	Vuln. Area	States Exam.	Filter States	Tested States	Succ. States	Loss States	Comp. Time	Err.
1	1.0	1	4	0.010	54322	53669	645	641	4	240.27	2
1	1.0	1	9	0.010	243286	240554	2317	2708	5	1004.27	0
1	1.0	1	16	0.010	243286	240554	2713	2708	5	1004.27	0
1	1.5	1	9	0.032	202660	199914	2686	2668	18	979.90	9
1	1.5	1	4	0.030	25230	24762	454	443	11	168.70	24
1	1.5	1	16	0.028	840301	829568	10546	10514	32	4132.77	1
1	1.5	2	9	0.026	202137	199425	2660	2646	14	1077.35	7
1	1.5	3	9	0.019	204780	202038	2695	2685	10	1048.27	2
1	2.0	1	9	0.057	194046	1906878	3188	3156	32	964.31	17
2	1.0	1	4	0.019	51678	51045	618	608	10	209.70	6
2	1.0	1	9	0.021	229575	226896	2643	2629	14	746.80	1
2	1.0	1	16	0.021	898963	885373	13403	13385	19	3297.23	1
2	1.0	2	9	0.021	229575	226898	2643	2629	14	534.84	2
2	1.0	3	9	0.019	230628	227950	2659	2649	10	915.40	2
2	1.5	1	9	0.052	193607	190957	2596	2580	16	572.96	13
2	2.0	1	9	0.083	187420	184213	3011	2990	21	1048.46	20
2	1.25	1	9	0.033	272584	269607	2932	2918	14	724.30	2

Table 10: NIR for the ATTL1 Network for two different upper layer topologies

α	Topology 1	Topology 2
1	0.059	0.089
4	0.009	0.037
16	0.002	0.028



(a) (b) (c) (d) (e)

Figure 28: Gabriel Network. Lower layer network is shown with solid lines. Upper layer network is shown with dotted lines. (a) Gabriel Network, L2 Network #1, Threat Radius = 10, Redundancy Factor = 1, Network Test = 40% (b) Gabriel Network, L2 Network #1, Threat Radius = 12.5, Redundancy Factor = 1, Network Test = 40% (c) Gabriel Network, L2 Network #1, Threat Radius = 15, Redundancy Factor = 1, Network Test = 40% (d) Gabriel Network, L2 Network #2, Threat Radius = 10, Redundancy Factor = 1, Network Test = 40% (e) Gabriel Network, L2 Network #3, Threat Radius = 10, Redundancy Factor = 1, Network Test = 40%

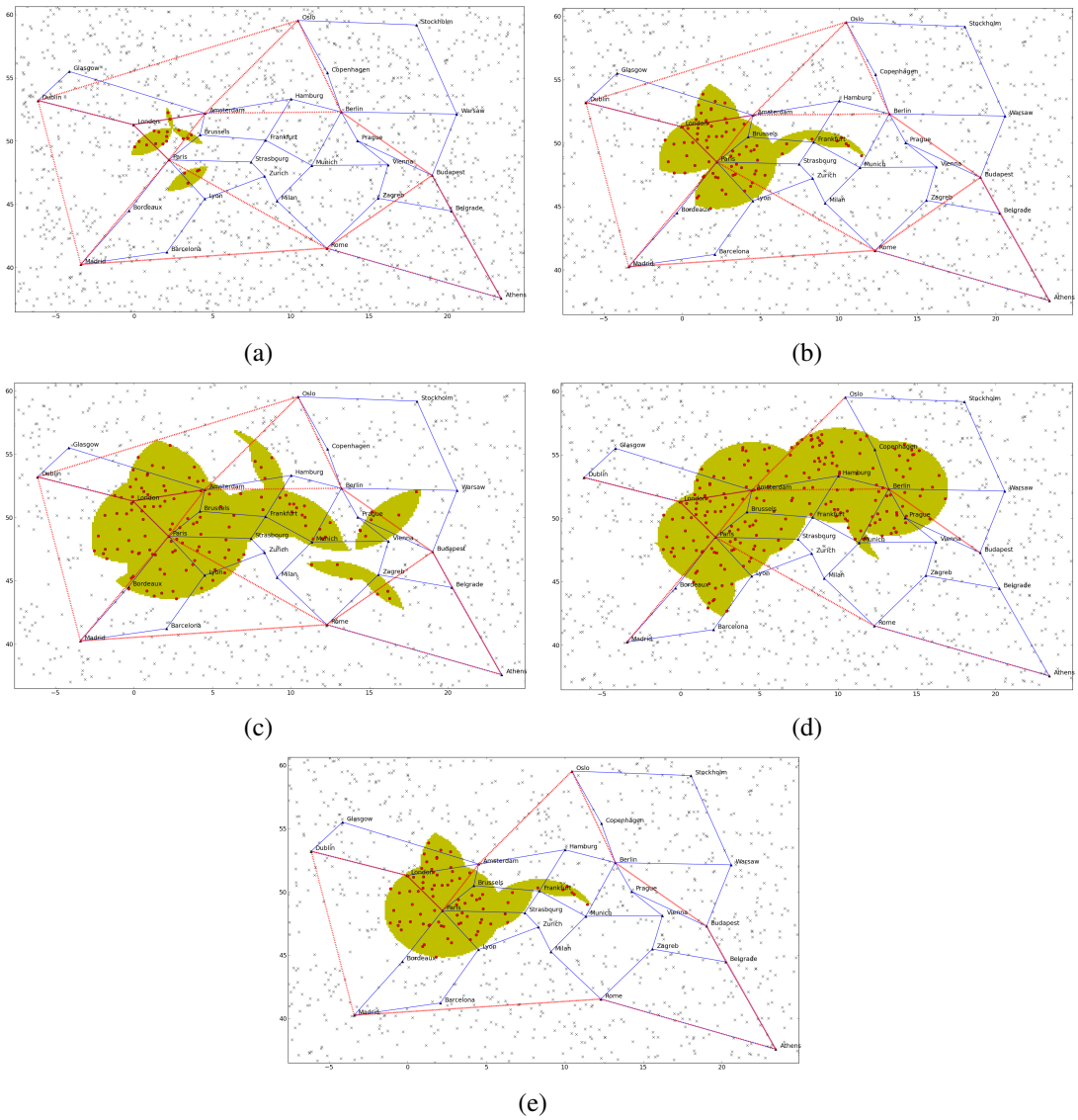


Figure 29: EU Network. Lower layer network is shown with solid lines. Upper layer network is shown with dotted lines. (a) EU Network, L2 Network #1, Threat Radius = 2.5, Redundancy Factor = 1, Network Test = 50% (b) EU Network, L2 Network #1, Threat Radius = 3.75, Redundancy Factor = 1, Network Test = 50% (c) EU Network, L2 Network #1, Threat Radius = 5.0, Redundancy Factor = 1, Network Test = 50% (d) EU Network, L2 Network #2, Threat Radius = 3.75, Redundancy Factor = 1, Network Test = 50% (e) EU Network, L2 Network #3, Threat Radius = 3.75, Redundancy Factor = 1, Network Test = 50%

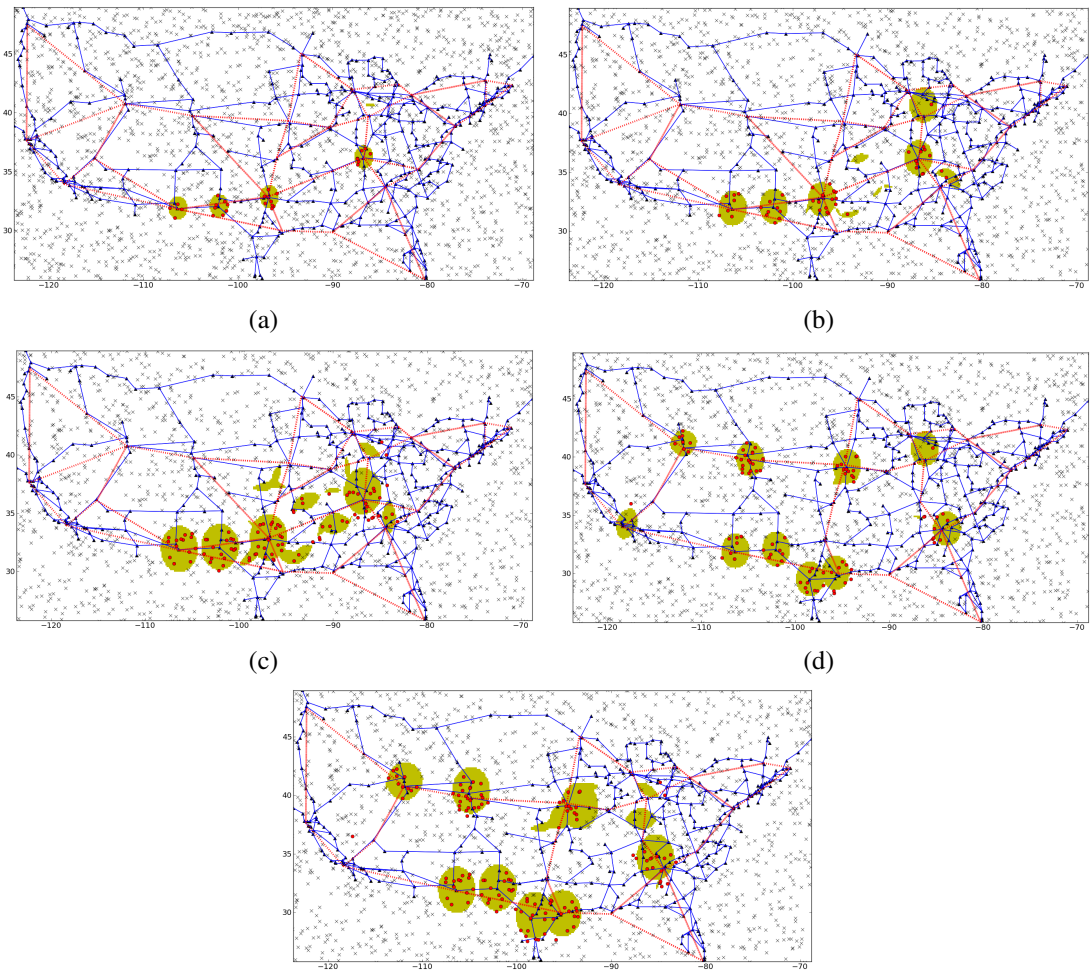


Figure 30: ATTL1 Network. Lower layer network is shown with solid lines. Upper layer network is shown with dotted lines. (a) ATTLA Network, L2 Topology #1, Threat Radius = 1.0, Redundancy Factor = 1, Network Test = 70% (b) ATTL1 Network, L2 Topology #1, Threat Radius = 1.5, Redundancy Factor = 1, Network Test = 70% (c) ATTL1 Network, L2 Topology #1, Threat Radius = 2.0, Redundancy Factor = 1, Network Test = 70% (d) ATTL1 Network, L2 Topology #2, Threat Radius = 1.5, Redundancy Factor = 1, Network Test = 70% (e) ATTL1 Network, L2 Topology #2, Threat Radius = 2.0, Redundancy Factor = 1, Network Test = 70%

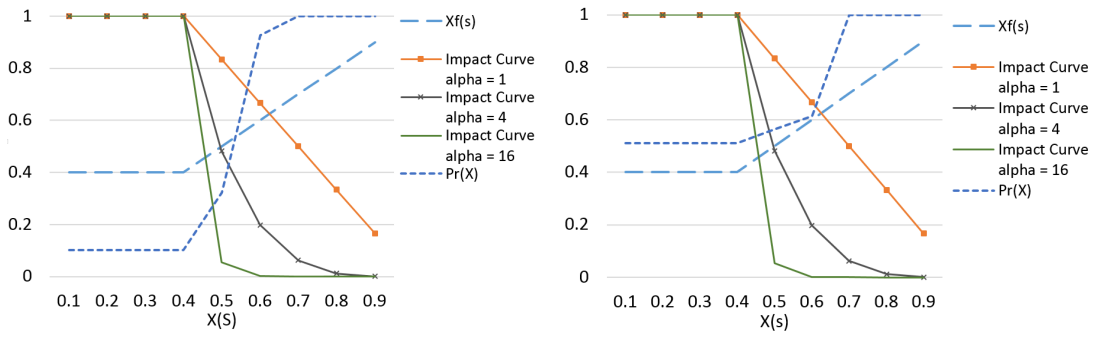


Figure 31: Gabriel Network NIR Impact and Probability Curves with Threat Radius of 15 (a) L2 Topology 1 (b) L2 Topology 3

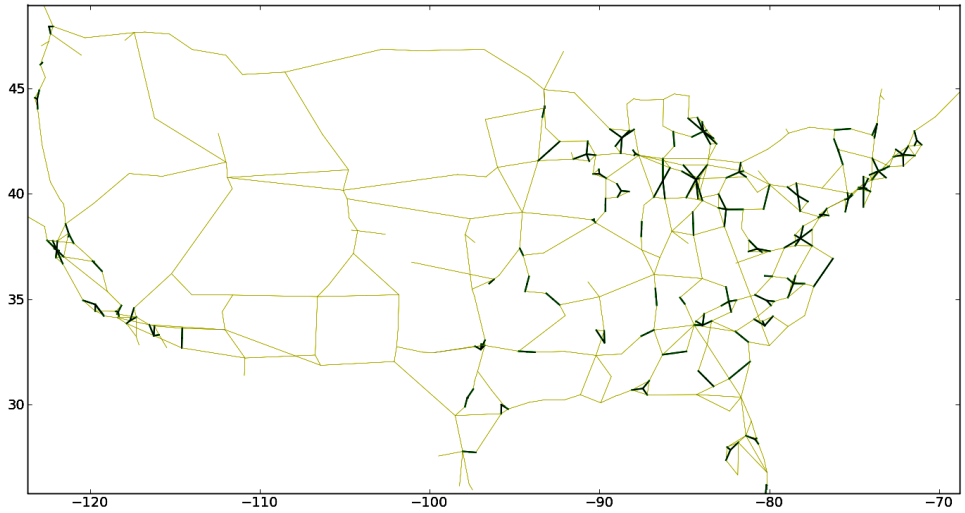
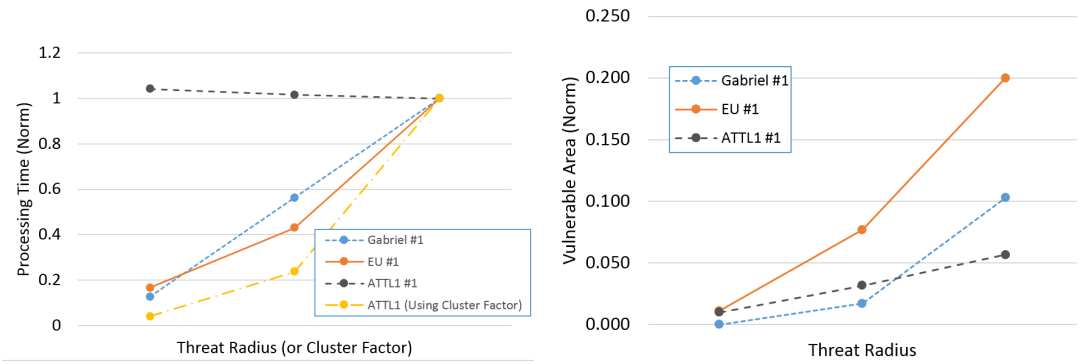


Figure 32: ATTL1 Network, Clustering with Threat Radius = 2.0, Redundancy Factor = 1, Network Test = 70%, Cluster location is at center of each tree (dark lines)

4.7.4 Performance of the Algorithm



(a) (b)

Figure 33: (a) Threat Radius/Cluster Size versus Processing Time (b) Threat Radius versus Vulnerable Area

In Figure 33(a), the effect that the threat radius has on processing time is shown normalized against the processing time at the largest threat radius. It is intuitive for the Gabriel and EU networks that the processing time rises as the threat radius increases. For the ATTL1 network, the processing time is relatively flat. This is because the complexity is controlled by the clustering algorithm. The plot of the clustering factor versus processing time is included to demonstrate this. In Figure 33(b), we see the vulnerable area versus the threat radius normalized against the total area in the network. The vulnerable area increases as would be expected when the threat radius is larger.

One of the concerns of using clustering for the purpose of reducing the number of states to be analyzed was the accuracy of the predicted vulnerability. In the Gabriel and EU networks (without clustering), we did not observe any errors. In the ATTL1 network, the effect of clustering on accuracy was shown in Table 9. As the number of clusters is reduced, errors do rise. However, even with 4 clusters the percentage of errors was still

less than 1% of 3000 random attacks.

4.8 Summary

During the SP-NSG process, one of the important benefits of state space pruning is that it can be used to find geographic vulnerabilities in networks. This benefit is exploited in this work. The NIR has the primary benefit that it can be used with different network tests, which allows the NIR to be configured specifically to the mission of the network and its applications.

However, for large networks, the vulnerability analysis can still become intractable. To maintain tractability in large networks, we also present a K-means clustering method. It relies on the idea of reducing the number of nodes for purposes of the state based analysis using clustering. Matrix transformation is used both for the K-Means Clustering Approach and the multilayer testing methods that maps geographic failures simultaneously onto multiple layers.

Small (16 node), medium (28 node), and large networks (383 nodes) with several different upper layer topologies are used to demonstrate our approach. Various threat radii were tested both for performance and impact on vulnerability. Typical results were seen with increased areas of vulnerability for larger threat radii. The performance was also reduced with larger threat radii. The notable exception to this is when other factors manage the growth in complexity like clustering are used. Maximum simultaneous node failures could also be used to manage complexity in a similar way.

Attacks were simulated to test the accuracy of the SP-NSG with clustering. Various clustering factors are tested for performance and accuracy. As the clustering factor is reduced, the number of nodes reduced to a single node is increased. This decreases accuracy but increases performance. Nine nodes in a threat radius seemed to provide good performance gains while maintaining reasonable accuracy. Different upper layer topologies were also tested. The selection of the topology had dramatic and sometimes unpredictable results. Loop topologies generally perform well with geographic attacks. Mesh topologies do not always perform as expected. This is related to vulnerabilities created by the initial provisioning algorithm.

The NIR was also demonstrated on the three test networks. It consistently followed our intuition on the prediction of resilient networks. We found success using the NIR with a high α value to predict resiliency. It would be interesting to learn more about the application of this metric with $\alpha = 1$ (or even less than one) and possibly its relationship with availability or reliability.

With respect to disaster planning, one factor that is not covered in this work is the communication of mission critical requirement information between the users and the network. In [59], extended Service Level Agreements (SLAs) are proposed for this purpose.

CHAPTER 5

PROVISIONING AND RESTORAL OF MISSION CRITICAL SERVICES FOR DISASTER RESILIENCE

During natural disasters such as earthquakes and hurricanes, providing communications is essential to support rescue and recovery operations associated with the disaster. This brings up a couple of important questions. *Is the network sufficiently prepared, from a capacity planning perspective, to handle the change in demands associated with an emergency event? When the event occurs, how can the network provide service to the most important users even when the network itself is challenged?*

In this chapter, we are proposing efficient methods to use Service Level Agreements (SLA) to improve capacity planning and provisioning for geographic emergency events as well as post-event restoration priorities. This includes new provisioning ILPs and heuristics, new restoration ILPs and heuristics, and suggested SLA extensions to support these algorithms [59]. A key finding of this work is that with only a small reduction in protection, significant cost savings are achievable.

This work is motivated by the idea that the network has different demands and requirements when it is *normal* than when it is in a *disaster* or *emergency* status. It is trivial to simply add capacity to support all demands needed during a normal mode and during an emergency event on primary and backup paths. Using *multi-situational* optimization

techniques, we can more efficiently manage that capacity to support *either* normal operations or emergency operations during geographic events but not necessarily both simultaneously in a stressed network. Once the emergency event occurs, and the network enters an emergency mode, the demands needed by the emergency response personnel become more important than in a normal mode. Ensuring that these demands are provisioned on the network *first* is the responsibility of the rerouting algorithms presented here.

An example of an emergency service that would need to be survivable during a disaster might be first responder communications or air traffic control needed to restore flights to a disaster area. Services, which under normal circumstances may be highly available but are not under disaster conditions, may include businesses that are not directly involved in disaster relief. Services that may need to be highly available during normal and disaster conditions may include hospitals.

SLAs can contain information about the importance of the service as well as the necessary service level objectives (SLO) for the service. We can both provision the network with the information in the SLAs as well as remediate the most important services based on this information. An augmented SLA structure is proposed to support initial network provisioning, and critical service remediation and rerouting during geographic network challenges.

5.1 Related Work

Although there is a large body of work related to assembling selected services for web service compositions based on QoS parameters in SLAs, we are not aware of any

work that attempts to provision services or demands across a network using SLAs that have the ability to provide resiliency from geographic challenges in the network. With respect to provisioning based on SLAs, most of the research is related to composing web services to meet a particular QoS level as proposed in [26] and [18]. These works assume that the number of web services to choose from are large and therefore, are a difficult problem. Our work is more interested in networks that have a relatively smaller number of emergency demands and present a different problem.

Several works address the use of SLAs. In [20], SLAs in the service-oriented architecture (SOA) environments are described as well as the web services agreements. They highlight a couple of important points including the need for SLAs to be machine readable and decomposable. By decomposable, they refer to the concept that different parts of the SLA may be needed by different entities for implementation. These features are also important to our work.

In networks supporting emergency services, the network typically has a strong role to play due to the high availability and sometimes low latency requirements that are imposed by emergency services. In [11], Badidi describes a brokered SLA service for an SOA that could be used by networks supporting emergency services to provide and implement QoS. The author provides an extensive discussion about how to implement service level objectives (SLOs). One of the advantages of brokered SLA services is that during events, the network has the ability to use the SLA to restore the most important services.

In [15], methods of restoration using SLAs during failures are discussed. The

goal is to ensure services are restored at a minimum QoS level during the event, provided services are searched to locate services that meet the minimum QoS. We try to account for this by provisioning using SLA QoS parameters. Designing for SLAs based on different quality of services for failures from normal operating conditions and restoration priorities have also been addressed and studied in [100, 103, 127].

Providing resilience to geographic challenges generally requires diverse paths between users in networks. In [32], Cheng et. al. proposed algorithms, including the iWPSP algorithm, that can be used to create diverse paths in networks. We used their algorithms in this work to ensure that the topology prior to provisioning has sufficient geographic diversity to support routing around many geographic challenges. For an early work on diverse routing, see [98].

Perhaps the most interesting work that has been done in the capacity planning area, with demands that have a variety of protection requirements, was proposed originally by Gerstel and Sasaki in [64] and is known as *Quality of Protection*. This was extended by Kuperman, Modiano, and Narula-Tam in [83] to include capacity planning using optimization. In this work, node-link formulations are used to provide capacity for working paths and demand specific backup path protection on a link by link basic. Since node-link formulations are used, paths do not need to be specified. In [142], varied levels of protection for network traffic engineering was modeled using a link-path formulation. Our work differs from these works in two ways. First, we use multi-situational link-path formulations to allow normal mode and emergency mode capacity planning that is not simultaneously needed. Second, by providing path generation, we are able to control the

distance between paths and therefore, provide protection against certain larger geographic events that may not have been possible using a node-link formulation.

In [121], extensive treatment is given to the construction of linear (or mixed-integer linear) programs that are capable of provisioning bandwidth in networks with constraints like path diversity and redundancy. Also in [121], multi-hour and multi-situational ILP approaches have been presented. We borrow from those ideas in this work. By treating the emergency demands as a separate provisioning situation, we are able to ensure that the links are provisioned to handle both the *normal* and *emergency* situations.

5.2 Mission Critical Disaster Resilience: Concepts and Illustration

Prior to any disaster, a network operates in the *normal* mode when the normal SLAs for the demands on the network are in force. Provisioning those demands would support these SLAs. Once a disaster occurs, a new paradigm comes into existence. Typically, emergency responders begin to use the network with different demands that usually have much more stringent requirements for availability and possibly latency. Network provisioning should allow for normal demands *or* emergency demands to be met. Additionally, when the event occurs, rerouting algorithms should be used to more strongly support the emergency services during the emergency event. It is important to note that during the emergency event, the network is challenged eliminating many of the primary paths available. For this reason, the rerouting algorithm only requires one path for all demands. The following steps are used to implement these concepts:

1. Collect *Normal* demands/SLAs

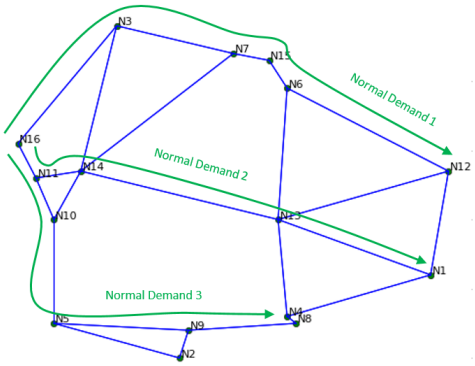


Figure 34: Gabriel Network with Sample Demands in Normal Operating Mode with no Redundant Capacity Provisioned on Diverse Paths. Redundancy can be Specified in Normal Operating Mode

2. Collect *Emergency* demands/SLAs

3. Select a resilient topology (not the subject of this work)

4. Assign capacity using a multi-situational capacity planning approach

5. Create a prioritized restoration approach for an emergency mode

To illustrate these ideas, Figures 34 and 35 show the normal and emergency operating mode concepts. If the normal demands were a volume of 10, then the links along the paths shown in Fig. 34 would need to have capacity added. To provision the Emergency Operating Mode, consider Fig. 35 to confirm that sufficient capacity exists (from the Normal Operating Mode) to support the Emergency Operating Mode demands. Links N4-N1 and N1-N12 would need additional capacity provisioned to support the Emergency Operating Mode. At this point, capacity should exist to support both configurations.

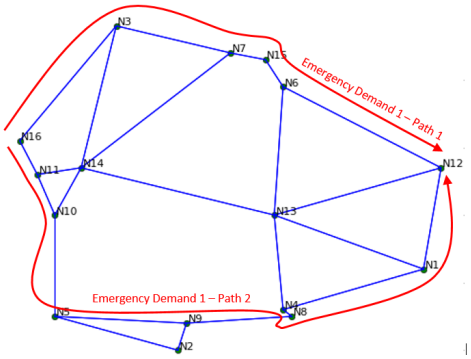


Figure 35: Gabriel Network with a Sample Demand in Emergency Operating Mode with 100% Redundancy Always Specified on a Diverse Path

5.3 Resilient Service Level Agreement (SLA) Configuration

In this work, we assume that SLAs can be customized to include new service level objectives (SLOs).

In addition to typical information included in the SLAs that are used to complete capacity planning, many common SLAs include specifications of service availability and system response times. In the WS-Agreement specification [9], the concept of a business value list is described as the ability to assign importance, penalties, and rewards to the SLOs.

In addition to other common SLOs, we propose specifying the following SLOs for use in this work: response time, availability, and survivability.

5.3.1 Response Time.

This is the maximum end-to-end response time that can be tolerated by a particular demand or service (T_d). This is used to compute the maximum utilization on the links. If

the path length per demand d using path p is L_{dp}, the maximum delay per link is assumed to be the worst case average response time divided by the path length for each demand that passes through that link:

$$W_{max} = \max((T_d)/L_{dp}). \tag{5.1}$$

Using $M/M/1$ approximations, we assume that the link delay, W, is as shown in (5.2) where τ is the average service time and ρ is the utilization:

$$W = \frac{\tau}{1-\rho}. \tag{5.2}$$

Rearranging, the maximum utilization per link is given by

$$\rho_e = 1 - \frac{\tau_e}{W_{max}}. \tag{5.3}$$

where τ_e is the average service time on link e. Note that a utilization model based on different response time requirements could easily be implemented.

5.3.2 Availability.

This is used to specify the service availability. If a given network availability is A_N and the required demand availability is A_d, the number of flows carrying the same information is increased until the availability offered is greater than A_d. Here, we assume that the flows are independent. If the number of flows is represented by the *redundancy factor* χ_d, then the relation with the availability can be stated as:

$$A_d = 1 - (1 - A_N)^{\chi_d}. \tag{5.4}$$

Note that the availability cannot be improved without path diversity (see the path diversity mechanisms discussed further in Section 5.4.1. Additional redundancy over the demand is provisioned on diverse paths to support availability. From (5.4), we can determine the requirement on the redundancy factor χ_d as follows:

$$\chi_d = \frac{\ln(1 - A_d)}{\ln(1 - A_n)}. \tag{5.5}$$

5.3.3 Survivability.

Survivability (S_d) is specified for the *emergency* demands d that are required during times of network challenge. These demands will be provisioned separately to ensure that they can be rerouted during network challenges. They will also be rerouted with priority during network challenges. It is important to note that if a service with survivability is specified but is not of high availability, it may not be restored during a *normal* component failure but would be rerouted with priority during network challenges.

5.3.4 Example of Resilient SLA Configuration

To illustrate the mapping of an SLA to the geo-diverse provisioning and rerouting models, consider the following example to show sample network calculations for maximum utilization and redundancy factor.

- Calculating Utilization
  - Max allow delay per link, $W_{max} = 50$ ms
  - Average Service time, $\tau_e = 15$ ms

- Then, from (5.3), we have $\rho_e = 0.7$

- Calculating Redundancy Factor

  - Network Availability, $A_N = 0.99$

  - Desired Demand Availability, $A_d = 0.999$

  - Then from (5.5), $\chi_d = 1.5$

5.4 Geo-Diverse Provisioning Models

We use two different approaches that add capacity with the goal of resiliency during *unknown* geographic events. The first one is a multi-situational mixed-integer linear programming (MILP) optimization with a link-path formulation utilizing a geo-diverse path generation method. The second one is a provisioning heuristic that adds capacity to the network along a set of pre-determined diverse paths for each demand. The advantage of these approaches is that we can generate the minimal number of paths that achieve the level of geo-diversity required. In this work, the number of paths is limited to 4. Before we do that, we briefly describe our path generation scheme.

5.4.1 Geo-Diverse Path Generation

In [32], Cheng et al. presented and analyzed two heuristics for generating geographically diverse paths between a source and destination in networks. The first heuristic is the Iterative WayPoint Shortest Path (iWPSP). The second heuristic is the Modified Link Weight Heuristic (MLW). In this work, we use iWPSP to create paths for this work.

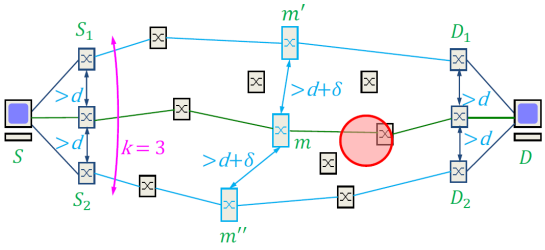


Figure 36: Iterative Waypoint Shortest Path Heuristic [32]

Briefly, the goal of iWPSP is to select a set of waypoint nodes from the source to destination that are d distance apart denoted as S_i and D_i. The number of paths determines the size of the set selected. Afterwards, another set of $d + \delta$ separated nodes are selected $(m, m', m'', ...)$ in the center of the source to destination path, Dijkstra's shortest path algorithm is used to select paths between the first set of nodes after the source and the set in the center of the path. This is repeated for the center to the destination sets of nodes as shown in Fig. 36. For more details, refer to [32].

5.4.2 Geo-Diverse MILP Provisioning

The geo-diverse MILP formulation shown is a novel use of the multi-situational link-path formulation from [121] that contains constraints for path diversity, additional capacity for redundancy and a set of constraints specifically designed to augment the network to account for emergency services. The link-path formulation requires pre-calculation of the routing paths to be potentially used for each demand—these paths are determined as briefly discussed in Section 5.4.1.

Table 11: Geo-Diverse Provisioning Notation

constants

δ_{edp}	1 if link e belongs to a path p realizing demand d, 0 otherwise
h_d	volume of demand d
χ_d	redundancy factor of demand d
S_d	survivability factor for demand d : '1' if demand d is needed for emergency mode, else '0'
ζ_e	cost for link bandwidth y_e
ξ_e	initial cost of link based on distance of e
ρ_e	maximum utilization on link e
H	conversion factor for binary variable

variables

x_{dp}	normal flow allocated to path p of demand d
w_{dp}	emergency flow allocated to path p of demand d
y_e	flow on link e
u_e	'1' if link e is used, else '0'

minimize

$$F = \sum_e \zeta_e y_e + \xi_e u_e \tag{5.6}$$

subject to

$$\sum_p x_{dp} = h_d \chi_d \tag{5.7}$$

$$\sum_d \sum_p \delta_{edp} x_{dp} \leq \rho_e y_e \tag{5.8}$$

$$x_{dp} \leq h_d \tag{5.9}$$

$$y_e \leq H u_e \tag{5.10}$$

$$\sum_p w_{dp} = 2 S_d h_d \tag{5.11}$$

$$\sum_d \sum_p \delta_{edp} w_{dp} \leq \rho_e y_e \tag{5.12}$$

$$w_{dp} \leq h_d \tag{5.13}$$

$$x_{dp} \geq 0 \tag{5.14}$$

$$y_e \geq 0 \tag{5.15}$$

The above MILP formulation is used to provision for the normal operating mode demands as well as for the emergency demands for survivability. The notations used in this formulation are summarized in Table 11. The objective shown in (5.6) minimizes the cost per link bandwidth assigned plus the initial link cost based on the link distance. (5.7) maps demand h_d to the flow paths. (5.8) maps the flow paths to the network links with additional capacity as needed to support the latency requirements (from (5.3)). (5.9) requires that no more than the volume of a demand be allowed on a given path. This forces any redundant capacity to be provisioned on other paths. In (5.10), if H is larger than $\max(y_e)$, then binary variable u_e will take on a '1' if y_e has capacity provisioned and '0' otherwise. Practically, H can be just a large number.

The additional constraints shown in (5.11) - (5.13) are used to provide additional capacity to support the Emergency Operating Mode demands. (5.11) maps demand h_d with 100% redundant capacity to the Emergency Operating Mode flow paths w_d; hence, the multiplier 2 is used. (5.12) maps the flow paths to the network links with additional capacity as needed to support latency requirements (from (5.3)). (5.13) requires that no more than the volume of a demand be allowed on a given path. This forces the redundant capacity to be provisioned on other paths. The remaining constraints are used for defining the requirements on all the variables.

5.4.3 Geo-Diverse Heuristic Provisioning

To reduce the complexity of solving the MILP model in the provisioning process, we also propose a heuristic to assign the capacity to the generated paths based on the

demands. In Algorithm 8, capacity is assigned to each path based on the length of the path. The shortest path will always receive the full volume of the demand. The next shortest path will receive the redundant capacity for that demand $(\chi_d - 1)h_d$. If the demand is marked for Emergency Operating Mode, the next shortest path will also receive the full volume of the demand. The additional capacity to account for latency requirements is added as the paths are mapped to the links. This heuristic approach is not intended to duplicate the MILP approach but rather used as an approximation.

Algorithm 8 Geo-Diverse Heuristic Provisioning

for all d in D **do**
  $x_{d1} \leftarrow h_d$
  if $S_d = 1$ **then**
    $x_{d2} \leftarrow h_d$
  else
    $x_{d2} \leftarrow (\chi_d - 1)h_d$
  end if
  for all p in 2 paths generated for demand d **do**
    for all e in E links in path p for demand d **do**
      $y_e \leftarrow y_e + x_{dp}/\rho_e$
    end for
  end for
end for
for all e in E **do**
  if $y_e > 0$ **then**
    $u_e = 1$
  else
    $u_e = 0$
  end if
end for
Compute cost function F given by (5.6)

Table 12: Geo-Diverse Rerouting Notation

constants

δ_{edp}	1 if link e belongs to a path realizing demand d, 0 otherwise
h_d	volume of demand d
n_d	diversity factor for demand d
S_d	survivability factor
c_e	Capacity available on link e after an event
ρ_e	maximum utilization on link
α_d	incremental normalized cost factor($\alpha_d \leq S_d$)

variables

x_{dp}	flow allocated to path p of demand d
μ_d	1 if demand d is provisioned

5.5 Post-Event Service Rerouting

After a geographic event occurs causing potentially widespread outages, the initial activity is rerouting of services. A common method is to reroute the emergency operating mode services first, and then reroute the remaining services using shortest path routing. If a path with the necessary link capacity does not exist, that demand is *skipped*. The link capacity on the found path is decreased by the demand volume. We refer to this as heuristic rerouting.

We propose an ILP model here to select demands to reroute based on a number of factors including the capacity remaining in the network, paths available for that service, priority (translated into cost) of that demand.

Geo-Diverse Rerouting Formulation

maximize

$$F = \sum_{d} (\alpha_d + S_d)\mu_d \tag{5.18}$$

subject to

$$\sum_{p} x_{dp} = h_d \mu_d \tag{5.19}$$

$$\sum_{d} \sum_{p} \delta_{edp} x_{dp} \leq c_e \rho_e \tag{5.20}$$

$$x_{dp} \geq 0 \tag{5.21}$$

$$\mu_d \text{ binary} \tag{5.22}$$

The notations used in the geo-diverse rerouting formulation are summarized in Table 12. The objective (5.18) maximizes the number of demands that are filled if the demands have higher weights in terms of the survivability factor and incremental normalized cost factor (α_d) for that demand, where the demands selection is indicated by the binary variable μ_d. This ensures that the Emergency Operating Mode demands are rerouted with high priority and then the remaining demands rerouted as capacity exists. To ensure the Emergency Operating Mode demands are rerouted with high priority, α_d would need to be a fraction of 1 (value of S_d for an emergency demand). Typically, we use $\alpha_d = 0.1$. (5.19) and (5.20) are constraints to assign demand to link capacity for the demands selected. Notice that c_e is the available link capacity increased to account for maximum link utilization to achieve the response time requirements *after* the event has occurred. The goal is to maximize the emergency demands that can be fulfilled in an

optimal manner.

The other important aspect of the rerouting formulation is the relaxation of several key constraints found in the provisioning formulation. These include the requirement for redundant capacity and the requirement for multiple path availability for demands. During remediation and rerouting, the goal is that the most important demands be rerouted regardless of the diversity or redundancy that is available on the network.

5.5.1 Model Implementation

The provisioning and rerouting heuristics were implemented using the Python language on a machine with an Intel I7 processor with 8Gb RAM. Python was also used to create the text based MILP files needed for the provisioning and rerouting MILP formulations. The LPSolve [19] optimization program, was called from Python to solve the MILP and ILP formulations. Random numbers needed for the attack analysis were generated using the Python Random library.

From [121], we know that solving linear problems using the Simplex algorithm is exponential time algorithm in the worst case, but in practice is generally O(n+m), where n is the number of variables and m is the number of equations.

The MILP provisioning problem was solved relatively quickly in all cases with the worst case solution time in the larger Nobel-eu network (28 nodes, 82 links, and 780 demands) being 78 seconds. Because of the relatively quick solution time, measures were not taken to reduce the solution time like relaxing the binary variables. However, for the

rerouting problem, there were attacks that generated MILP problems that were not solvable in a reasonable time frame. In those instances, the binary variables μ_d were relaxed to $0 \leq \mu_d \leq 1.0$. This allowed a feasible solution to be created with a linear program. In the relatively rare situation that a μ_d variable did not solve to 1.0 or 0.0, it was assumed that the i link would not support the traffic. This is a conservative assumptions that would results in slightly more demands being able to be provisioned than are calculated here.

5.6 Results

We implemented the provisioning and rerouting heuristics in the Python language on a machine with an Intel I7 processor with 8Gb RAM. Python was also used to create the text based MILP files needed for the provisioning and rerouting MILP formulations. The LPSolve [19] optimization program, was called from Python to solve the MILP and ILP formulations. Random numbers needed for the attack analysis were generated using the Python Random library.

To test the capabilities of both the geo-diverse provisioning and rerouting algorithms, we evaluated the models on two network topologies. Fig. 34 shows a Gabriel network. The Gabriel network was used because it provides a good grid like structure with a configurable number of nodes (16 nodes). The capabilities of Gabriel networks are studied in [28]. The second network, shown in Fig. 37 is from the medium sized Nobel-eu network that is available at the Survivable Network Design Library (SNDlib 1.0) [119].

We first briefly comment on computing time and run time. We found that the MILP provisioning problem was solved relatively quickly in all cases with the worst case

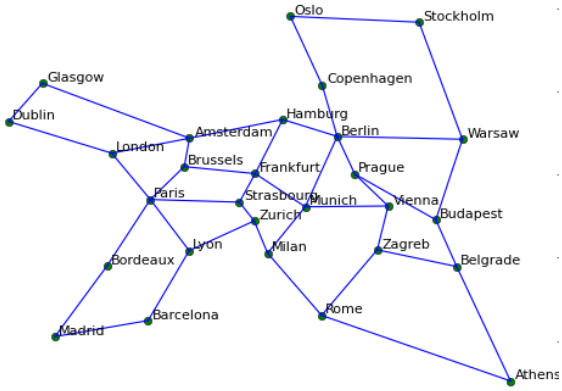


Figure 37: Nobel-eu Network

solution time in the larger Nobel-eu network (28 nodes, 82 links, and 780 demands) being 78 seconds. Because of the relatively quick solution time, measures were not taken to reduce the solution time like relaxing the binary variables. However, for the rerouting problem, there were attacks that generated MILP problems that were not solvable in a reasonable time frame. In these instances, binary variables μ_d were relaxed to $0 \leq \mu_d \leq 1.0$. This allowed a feasible solution to be created with the relaxed linear program. This is a conservative assumption that would result in slightly more demands being able to be provisioned than are calculated here.

For each network, six scenarios were tested as shown in Table 13. For all scenarios, the demand structure was created by fully connecting all nodes in the network to each other for a total of 240 and 756 demands, respectively, for the Gabriel and Nobel-eu networks each with a volume of 1. The demands marked for the Emergency Operating Mode were randomly selected based on the percentage in Table 13. The redundancy χ_d was set according to Table 13 and the appropriate provisioning algorithm was executed.

Table 13: Scenario 1-6 Configuration (Gabriel and Nobel-eu Networks)

#	Provision/ Reroute Alg.	Redundancy Factor (χ_d)	% Emer. Demands ($\%S_d = 1$)	Threat Radius Gabe/Nobel
1	ILP/ILP	1.0-2.0	25%	10/5
2	ILP/Heur	1.0-2.0	25%	10/5
3	Heur/ILP	1.0-2.0	25%	10/5
4	Heur/Heur	1.0-2.0	25%	10/5
5	ILP/ILP	1.0	25%	5-15/3-7
6	ILP/Heur	1.0	10%-50%	10/5

ρ is set to 0.7 as was calculated in the earlier SLA example.

After all demands were provisioned on the networks, a series of 100 attacks were generated at random locations in the network. Each attack failed all links that intersected with a circle surrounding the attack known as the *threat radius* representing the estimated radius of a disaster. The rerouting algorithm was then executed and the number of demands that could be rerouted successfully were computed without violating the capacity of the links as provisioned. The percentage of successfully rerouted demands wss the primary metric used to evaluate the attempted remediation activity. Additionally, the cost to provision the network was computed. In this analysis, if the demand source or demand destination nodes were contained in the threat radius, that demand was considered to have failed. In some cases, that would make the demand restoration percentages seem low.

5.6.1 Provisioning Costs

Figure 38 shows the provisioned Gabriel network using MILP Provisioning. Also shown are the attack locations used to test the network. The width of the lines indicates

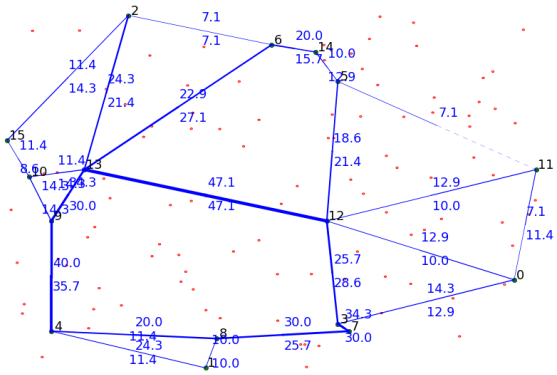


Figure 38: Gabriel network with MILP provisioning ($\chi_d = 1.0$) and attack locations. Line width indicates link capacity provisioned in each direction

the capacity of the link in that direction. The actual capacity is also shown in the text next to the link number. Dashed lines indicate no capacity assigned. Path generation uses the iWPSP algorithm.

The provisioned Nobel-eu network is shown in Fig. 39. The influence of the distance component of the provisioning cost tended to drive capacity toward the shortest paths in the center of the network in both Gabriel and Nobel-eu networks. The MILP formulation is meant to drive just enough capacity to the edges of the network to cover contingency scenarios when the center of the network is lost.

The cost of each of the provisioning methods is shown in Fig. 40(a) and Fig. 41(a). The methods include MILP and Heuristic provisioning methods using iWPSP path generation. The cost includes a fixed cost for each link used based on the length of the link plus a cost related to the amount of capacity assigned to that link. The cost factors for each network were adjusted so the initial costs and capacity related costs are approximately equal.

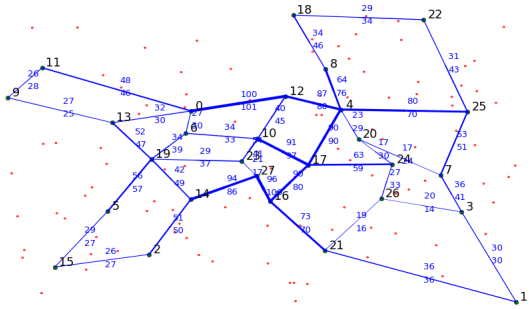


Figure 39: Nobel-eu Network with MILP Provisioning ($\chi_d = 1.0$) and attack locations. Line width indicates link capacity provisioned in each direction

The Heuristic Provisioning methods were always slightly more expensive than the MILP methods. Note that in the heuristic approach, the redundant capacity is always placed on path 2. On the other hand, in the MILP model, multiple paths are available for certain demands (as determined by the iWPSP algorithm). This enable the paths to be chosen to minimize the cost function.

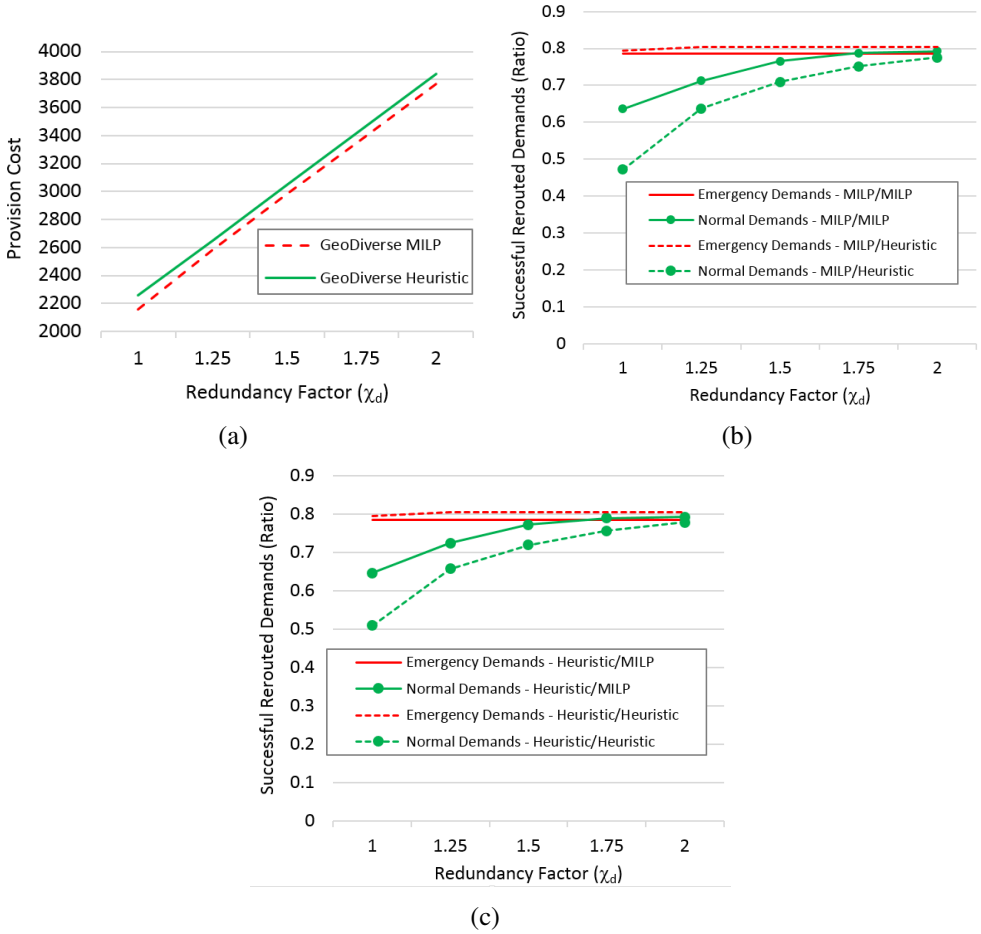


Figure 40: Gabriel Network Performance (a) Provisioning Cost (b) Scenarios 1&2, MILP Provisioning (c) Scenarios 3&4 - Heuristic Provisioning

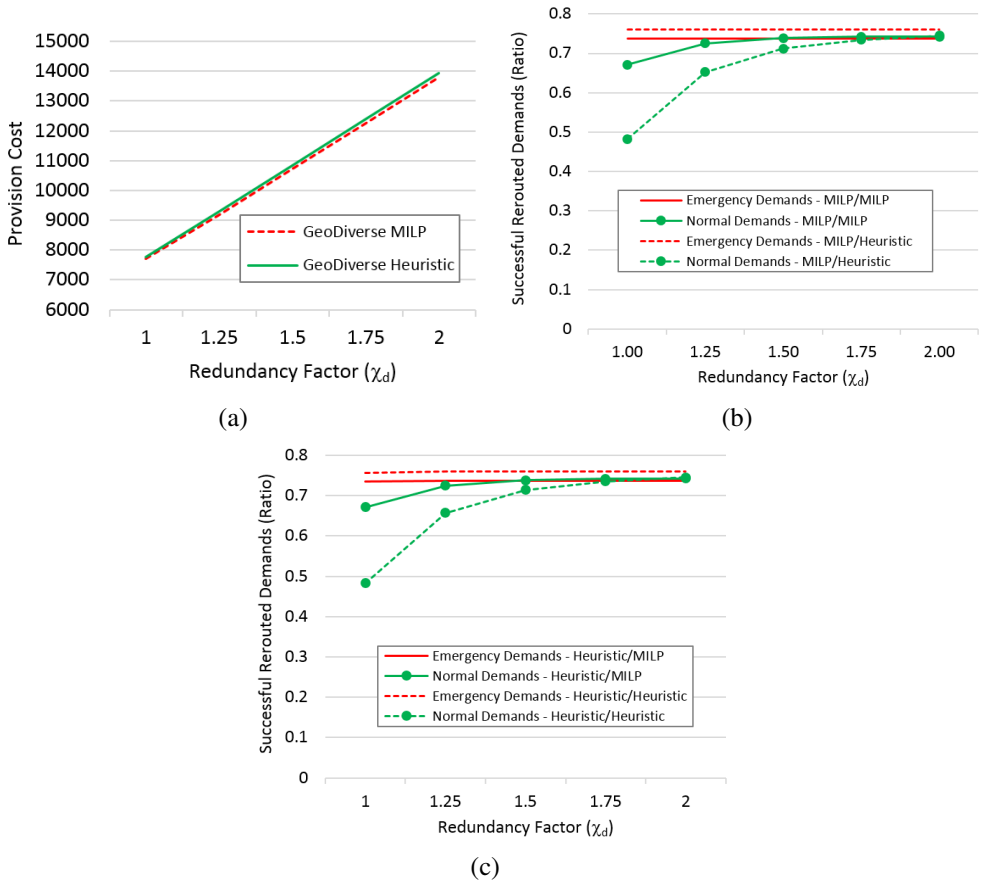


Figure 41: (a) Provisioning Cost (b) Scenario 1&2, MILP Provisioning (c) Scenarios 3&4 - Heuristic Provisioning

5.6.2 Rerouting Performance

The average percentage of successful demand rerouting in the Gabriel network is shown in Fig. 40(b) and Fig. 40(c). Figures 41(b) and 41(c) show the Nobel-eu network. We observe that the heuristic approach to rerouting always provides a slightly better performance than the MILP restoration approach for the demands marked for emergency operation mode. However, the MILP significantly outperformed the heuristic approach for the remaining normal services. As expected, the difference between emergency services and normal demands is significantly more apparent when the redundant capacity associated with normal services (since emergency demands always have 100% redundancy) is near 1.0 (no redundancy). These results were not unexpected as the heuristic method was similar to a priority method.

Comparing the MILP and heuristic provisioning approaches, we see that the rerouting performance is almost identical. The costs for the heuristic approaches were higher for the Gabriel topology and slightly higher for the Nobel-eu network. These two facts would indicate that generally, the heuristic approach is capable of creating competitive solutions with the MILP provisioning approach, albeit at a higher cost.

5.6.2.1 Effects of Threat Radius

Figures 42(a) and Fig. 42(c) show the effect of changing the threat radius on the performance of the rerouting algorithms. As would be expected, the performance decreases almost linearly based on the size of the threat radius. We observed that in general, the decrease in performance was linear with the increase in the threat radius.

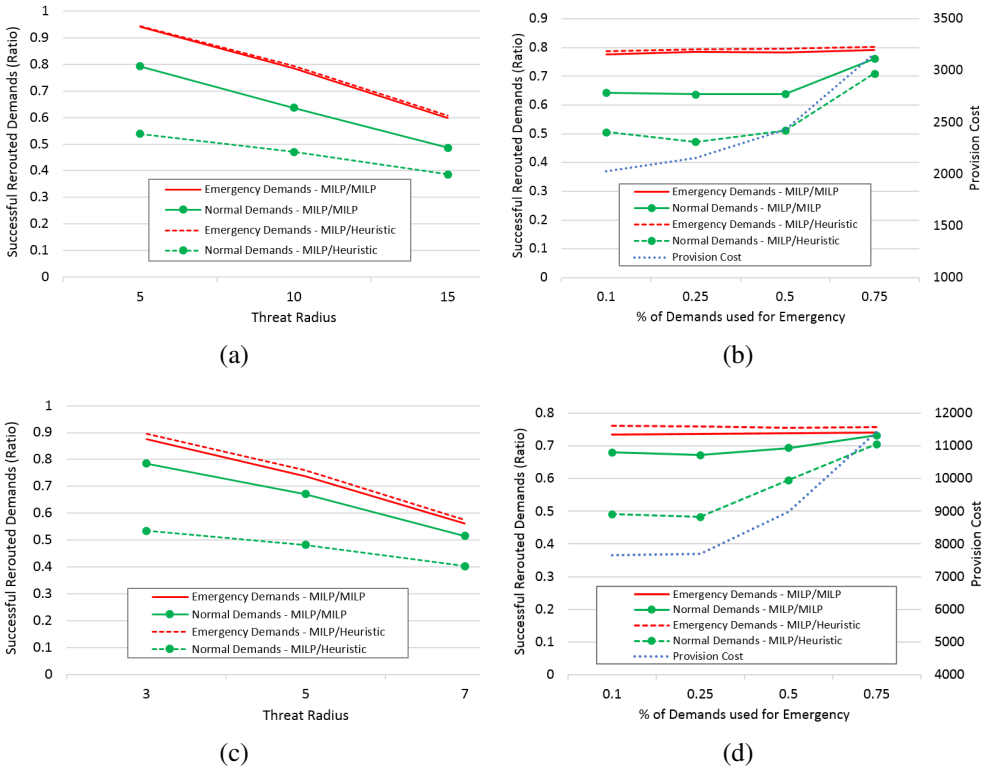


Figure 42: Network Performance (a) Gabriel Network, Scenario 5 (b) Gabriel Network, Scenario 6 (c) Nobel-eu Network Scenario 5 (d) Nobel-eu Network, Scenario 6

This is not an unexpected result. As we noted earlier, the MILP performance for the emergency demands was slightly less than the heuristic performance for the emergency demands. However, the MILP rerouting solution for normal demands was *significantly* better than the rerouting heuristic solution

5.6.2.2 Effects of Increasing the Ratio of Emergency Demands

Figures 42(b) and Fig. 42(d) help us to make a couple of observations. When the percentage of emergency demands is low ($< 25\%$ of all demands), the difference between the performance of emergency demands and normal demands is consistent. As

the percentage of emergency demands increases ($> 50\%$) the performance of the all of the demands increases since more demands are being provisioned with 100% redundancy at the increased costs as shown.

5.7 Summary

This work proposes a framework to use Service Level Agreements (SLAs) to provide much of the information necessary to solve the challenging problems of defending and remediation in network vulnerability to geographical events like natural disasters. We form the solutions to these problems as provisioning and rerouting solutions to prevent damage from geographic events. From the SLA parameters: system response time, availability, and survivability, we can create integer linear programs and heuristics to solve the provisioning and rerouting problems.

The MILP and heuristic approaches to the network provisioning problem with geographic challenges are both found to be effective solutions. The performance of both approaches are similar. The cost of the MILP approach varies from slightly less expensive to moderately less expensive than the heuristic solution.

The MILP and heuristic approaches to the rerouting problem have significant differences. The heuristic approach performs slightly better than the MILP solution for the emergency services. But, the normal services see significant benefit under the MILP rerouting. solution. The MILP formulation provides a better overall solution to the rerouting problem.

A final observation is the importance of diverse path generation. Once the provisioning is complete, the network is reliant on that process to help deal with the restoration during an event. If good path generation is an integral part of the provisioning process, the chance of successful restoration is significantly higher. The iWPSP path generation methods provided that capability for this work.

Our approach could be extended to multi-layer networks, service oriented architecture (SOA), and software defined networks (SDN). It is not uncommon for emergency applications to be deployed on an SOA, like the Federal Aviation Administration's System Wide Information Management (SWIM) network [143].

CHAPTER 6

TOPOLOGY IMPROVEMENTS TO AVOID HIGH IMPACT GEOGRAPHIC EVENTS

During a disaster, communications systems can be devastated. When this occurs, a communications *black hole* is created at the center of the disaster. This causes a significant lack of situational awareness immediately that extends throughout the geographic impact zone. The network infrastructure that remains becomes stressed from a flood of users trying to communicate. This is the basic description of a geospatial event in a network. Events of this nature are not limited to small or large geographic areas.

Typically network designers separate topology from geography. In this chapter we develop optimization models to add nodes to a network topology that eliminate the geographic vulnerabilities, protecting the network from geospatial events. We develop models based on Integer Linear Programming (ILP) and Particle Swarm Optimization (PSO) to optimize networks for both point-to-point and all-terminal applications. This work was published in [55]. To demonstrate the flexibility and applicability of both approaches, we use simple wireless propagation link models as well as more complex link models that include obstructions like buildings and other terrain features.

Both the ILP based approach and the PSO based approach were successful in mitigating geographic vulnerabilities found in our test networks. In Section 1.1, the geographic vulnerability of a network with nodes N, links E, and threat radius r is defined as

the *vulnerable area* of a network. We further define $V(N, E, r)$ as the Geographic Vulnerability which participates in the objective function of the PSO approach. The network test $V(N, E, r)$ used for this portion of the work is based on a connectivity model. But any network test could be used. Since the $V(N, E, r)$ is non-linear, the ILP approach is only an approximation. The PSO approach can use $V(N, E, r)$ as the objective providing a more direct solution. Another important difference between the two approaches is that the ILP approach can augment the network with multiple nodes concurrently, whereas the PSO approach adds one node at a time.

6.1 Related Work

There have been significant recent advances in the assessment of geographically correlated failures and network design related to geographically correlated vulnerabilities. Section 4.3 provides geographic vulnerability assessment methods used for this work.

Sen, Murthy, and Banerjee in [140] construct region disjoint paths between a source and destination. Their work has some similarities to our work. The regions in their work are predetermined. Li, Wang, and Jiang [89] expand on the work in [140] in two areas. First, they consider multiple regions as opposed to one. Second, they augment the network by adding link capacity to ensure all traffic can withstand multiple region failures. Instead of predetermining regions, they assume regions are centered at each node. Our work designs the network assuming the region can be centered anywhere in the network. Also, we augment our network with new node locations as opposed to finding region disjoint paths within existing nodes.

Alenazi, Cetinkaya, and Sterbenz recently proposed methods to augment network resilience by using algorithms that add network links that maximize the improvement in Algebraic Connectivity with the new link in [7] and maximize the improvement in the pairwise path diversity in [8].

In [30], Cetinkaya, Broyles, Dandekar, Srinivasan, and Sterbenz have taken a different approach and developed a simulation platform that is capable of locating geographic vulnerabilities in networks. Their framework is capable of modeling both malicious and non-malicious attacks, disasters, and wireless challenges.

6.2 Overview of Methods

In Section 1, geographic vulnerability, geospatial event, and threat radius are defined. We further define $V(N, E, r)$ as the Geographic Vulnerability. Figure 43 shows an example of geographic vulnerabilities in a 20 node network. Note that when multiple nodes need to be disabled concurrently, the vulnerable area is the intersection of the threat radius around those nodes. Algorithm 9 is used to compute $V(N, E, r)$.

Algorithm 9 Calculation of $V(N, E, r)$

$V(N, E, r) = []$
for all i in F Failure Modes (found with SP-NSG) **do**
  for all j in Nodes associated with i **do**
    if $j = 1$ **then**
      $V_i =$ Threat Radius(j)
    else
      $V_i = V_i \bigcap$ Threat Radius(j)
    end if
  end for
  $V(N, E, r) = V(N, E, r) \bigcup V_i$
end for

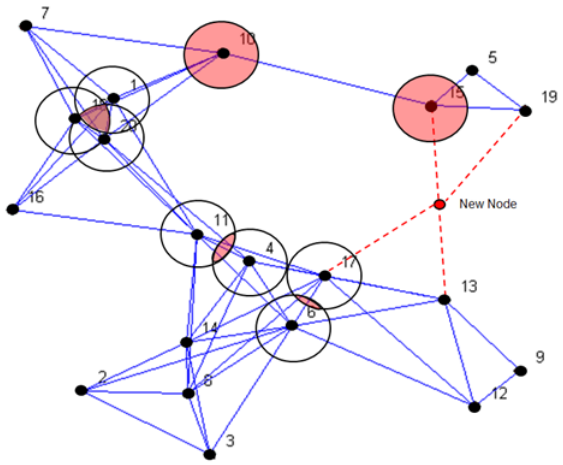


Figure 43: Wireless Network showing Geographic Vulnerabilities and a New Node Location that Eliminates the Vulnerabilities

With this in mind, we try to accomplish the following with this work:

1. Develop an Integer Linear Program (ILP) that augments a network by selecting locations to add nodes to a network which approximates the minimization of the geographic vulnerability $V(N, E, r)$ of that network. This applies both to point-to-point demands as well as point-to-any (all-terminal) demands. For point-to-point demands we create paths that are physically diverse from a line between the source and destination (s, t). New nodes are added (with a new node cost) that lower the diversity cost of the path. The all-terminal model is similar, using spanning trees as opposed to paths.

2. Develop a particle swarm optimization (PSO) approach to select a location to add a single node that minimizes the geographic vulnerability $V(N, E, r)$ for the point-to-point case and the all-terminal case. Because PSO is used, non-linear objective

functions can be employed. The objective function used is $V(N, E, r)$ as described in Appendix 1.

3. Test both approaches (ILP and PSO) with networks of various sizes (15 nodes to 50 nodes) and multiple link models, allowing our approaches to be used in wireless networks as well as wired networks. In addition, we include obstructions in the link models.

Since the $V(N, E, r)$ is a non-linear function, we chose two approaches. The ILP approach is a heuristic approach that attempts to minimize $V(N, E, r)$ by using weighting to create diverse paths. If the two paths are physically separated by at least the threat radius, the effect is to reduce $V(N, E, r)$ as shown in Figure 44. Both solutions are feasible. But, path 2 reduces $V(N, E, r)$ more than path 1 due to nodes 10 and 15.

The PSO approach directly uses the minimization of $V(N, E, r)$ as its objective function. PSO particles move across the network space using the PSO algorithm, testing the objective at each particle location as they move. The particle location that minimizes the objective function the most is added as a new node.

6.2.1 Integer Linear Program (ILP) Based Approach

The ILP formulation is based on the premise that the program is executed two times with opposite weighting each time, creating two diverse paths (point-to-point) or two diverse spanning trees (all-terminal). Since calculating geographic vulnerability $V(N, E, r)$ is an inherently non-linear problem, we use weights that make it more costly for the diverse paths or diverse spanning trees to be geographically close together. The *geographic*

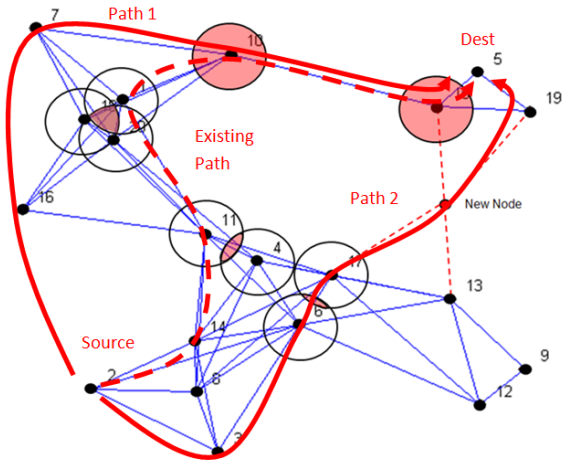


Figure 44: Point-to-Point Scenario Showing Two Potential Diverse Paths on Opposites Sides of the Network

space in which the network resides is weighted to push Path 1 or Spanning Tree 1 to a specific geographic area. The weights are then changed to push Path 2 or Spanning Tree 2 to the opposite geographic area. Possible new nodes are selected from a grid pattern with a *new node* cost to improve a poorly weighted path.

By ensuring that two paths exist through the network on opposite sides of the network, vulnerabilities located anywhere in the network can be mitigated. A single path may not mitigate the vulnerability, if it is located in the same area as the path is weighted toward as shown in Figure 2. The same rational applies for using spanning trees located in different physical areas of the network. Path 1 or Path 2 (Spanning Tree 1 or 2) may then be chosen to augment the network.

By ensuring that two paths exist through the network on opposite sides of the network, vulnerabilities located anywhere in the network can be mitigated. A single path

may not mitigate the vulnerability, if it is located in the same area as the path is weighted toward as shown in Figure 44. The same rational applies for using spanning trees located in different physical areas of the network. Path 1 or Path 2 (Spanning Tree 1 or 2) may then be chosen to augment the network.

Spanning trees are used in the all-terminal case ensuring that all nodes can communicate with all other nodes. We chose nodes with the lowest geographic diversity cost to serve as root nodes. This prevents poor root node choice from adding significant diversity cost to the optimization. Since, we are concerned only with connectivity, any spanning tree will suffice.

Figure 45 shows the node weights for an example point-to-point case. The weighting scheme is based on the perpendicular distance from a line across the geographic space that includes the source and destination nodes (s, t). For Path 1, the weights are normalized at zero on the furthest edge perpendicular to the source-destination line and grow toward the opposite side of the geographic space. For Path 2, the weights are reversed with zero being normalized to the opposite corner and growing the other direction. The effect is that Paths 1 and 2 would form on opposite sides of the network space geographically.

For the all-terminal example shown in Figure 46, the weighting scheme is based on an interior/exterior concept. For Spanning Tree 1, the weights start at the center at zero and grow toward the exterior of the space based on the distance from the center of the space. The weighting scheme for Spanning Tree 2 will have the furthest point in the space starting at zero and growing toward the center. Spanning Tree 1 will tend to form

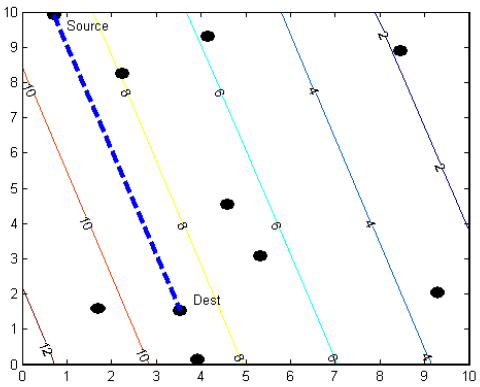


Figure 45: Point-to-Point Weighting Plan (Path 1) for an Example 10 Node Network

in the center of the network space and Spanning Tree 2 will tend to form on the edge of the graph providing diversity between the two trees.

$$W = d^{\alpha} \quad \alpha > 0 \tag{6.1}$$

Equation 6.1 shows the calculation of the geographic diversity cost W. α is a constant used to scale the weights exponentially. d is the distance from the (s,t) line (source - destination) or from the center for the all-terminal case. The ILP approach uses a node-link formulation as described in [121] that includes all nodes and links (existing and potential). The formulation follows.

minimize

$$F = \sum{}_n W_n v_n + C \sum{}_p v_p \quad n = 1 \cdots N \quad p = m \cdots N \text{ potential new nodes} \tag{6.2}$$

constraints (point-to-point)

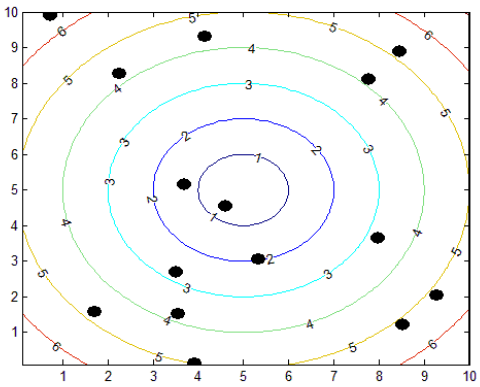


Figure 46: All-Terminal Weighting Plan (Spanning Tree 1) for an Example 15 Node Network $(\alpha = 1)$

$$\sum {}_e a_{en} u_e - \sum {}_e b_{en} u_e = \left\{ \begin{array}{ll} 1 & \text{if } n = s \\ 0 & \text{if } n \neq s, t \\ -1 & \text{if } n = t \end{array} \right\} \quad n = 1, 2, \cdots, N \tag{6.3}$$

$$\sum {}_e a_{en} u_e = v_n \quad n = 1, 2, \cdots, N \tag{6.4}$$

Equation 6.3 is the balance equation for a node-link formulation [121]. Equation 6.4 sets v_n to the units of flow passing through node n. In the point-to-point model, the total flow is 1, v_n is either 1 or 0.

constraints (all-terminal)

$$\sum {}_e a_{en} u_e - \sum {}_e b_{en} u_e = \left\{ \begin{array}{l} N - 1 \text{ if } n = root \\ -1 \text{ for all others} \end{array} \right\} \quad n = 1, 2, \cdots, N \tag{6.5}$$

$$\sum {}_e a_{en} u_e = v_n \quad n = 1, 2, \cdots, N \tag{6.6}$$

Table 14: Notations ILP Formulation

notation	
$e = 1, 2, \cdots, E$	List of all possible links
$n = 1, 2, \cdots, m-1, m, \cdots, N$	List of all existing$(1 \cdots (m-1))$ and potential new$(m \cdots N)$nodes
constants	
Wn	Diversity Cost at node n
a_{en}	Used for Node-Link formulation, 1 if node$n$ is the originating node of link e; else 0
b_{en}	Used for Node-Link formulation, 1 if node$n$ is the terminating node of link e; else 0
s	Source node of demand
t	Sink node of demand
N	Number of nodes (existing and possible)
C	Cost to add a node
K	Maximum number of possible added nodes (optional)
variables	
u_e	Binary variable is 1 if link e is used, else 0
v_n	Integer variable is indicative of traffic flows passing through the node.

$$y_i(t+1) = \left\{ \begin{array}{ll} y_i(t) & if\ f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & if\ f(x_i(t+1)) \leq f(y_i(t)) \end{array} \right\} \tag{6.7}$$

To compute $f(x_i(t+1)) = V(N, E, r)$ in a network of N nodes:

1. Convert location of particle $x_i(t+1)$ to new node
2. Add node to the set of nodes N
3. Generate set of links E based on set of nodes N.
4. Compute $V(N, E, r)$ as described in Algorithm 9

These stopping parameters halt PSO movement:

1. A geographic vulnerability threshold is reached.

Table 15: Test Network Configuration

Networks A 1-3 Configuration		
Nodes	15	
Wireless TX Level	15	dBm
Antenna Gain	0	dB
Receive Threshold	-98	dBm
Frequency (MHz)	2400	MHz
Link Generation Method	FSL	
Obstruction Loss	0	dB
Networks B 1-3 Configuration		
Nodes	25	
Wireless TX Level	25	dBm
Antenna Gain	10	dB
Receive Threshold	-120	dBm
Frequency (MHz)	1000	MHz
Link Generation Method	ITU-R	
Obstruction Loss	6	dB
Networks C 1-3 Configuration		
Nodes	50	
Wireless TX Level	18	dBm
Antenna Gain	10	dB
Receive Threshold	-120	dBm
Frequency (MHz)	1000	MHz
Link Generation Method	ITU-R	
Obstruction Loss	6	dB

2. Maximum iterations are reached.

6.2.2 Link Generation Techniques

To demonstrate the model usage in realistic scenarios, we tested different wireless link generation models that could include terrain obstructions. Two basic models were used to generate links.

1. Free Space Loss (FSL). Friis's Law is used to determine if sufficient power is available at the receiving node [112].

2. ITU-R Pedestrian Model with Obstructions. Equation 6.8 shows the ITU-R Pedestrian Model [112]. A 6 dB loss is added if the path crosses obstructions.

$$PL = 40log_{10}d + 30log_{10}f_c + 49; \tag{6.8}$$

6.3 Results

To evaluate the methods in this paper, we have selected 9 different randomly generated networks, three 15 node (A1 - A3), three 25 node (B1 - B3), and three 50 node (C1 - C3) networks. In networks A1-A3, we assumed no obstructions and the free space loss (FSL) link model. For networks B1-B3 and C1-C3, we chose to include obstructions and used the ITU-R Pedestrian Model to calculate loss. All networks are assumed to occupy a 10 km x 10 km space. The configuration parameters are shown in Table 15.

The ILP parameters needed to add nodes to a network are strongly dependent on the topology of that individual network. For the point-to-point case C was set to 1.0 and α was set to 0.5 for networks A(1-3) and B(1-3). For network C(1-3), α was varied from 0.25 and 1.0 till nodes were added to the solution. The same approach was used for the all-terminal case. In the high weight option, α ranges from 0.5 to 0.75. C remained 1.0. In the low weight option, C ranges from 2.0 to 5.0. The PSO approach used acceleration constants of $c1$ and $c2$ of 0.1 for all cases. Stopping parameters of 16 iterations and $V(N, E, r) = 0.03$ were used for all tests.

In Tables 16 and 17, we see the geographic vulnerability $V(N, E, r)$ of the A, B, and C networks with no augmentation, augmentation with the ILP approach, and augmentation with the PSO approach. It is interesting to note that the ILP approach can add multiple nodes. This occurs frequently because one node (in the predetermined location) may not sufficiently reduce the diversity cost to bring that solution into feasibility. Since the diversity cost is an approximation of $V(N, E, r)$, nodes can be added that reduce the diversity cost but do not improve $V(N, E, r)$. The ILP approach can be weighted such that nodes are cheap and many new nodes are brought into feasibility or such that nodes are expensive. This will create varying values of $V(N, E, r)$.

Table 16: $V(N, E, r)$ - Point-to-Point (added nodes shown in parentheses)

Network	Initial	ILP Optimized	Swarm Optimized
A1	0.182	0.028 (1)	0.028 (1)
A2	0.116	0.018 (1)	0.029 (1)
A3	0.063	0.000 (1)	0.009 (1)
B1	0.183	0.000 (1)	0.000 (1)
B2	0.333	0.000 (1)	0.000 (1)
B3	0.216	0.000 (1)	0.043 (1)
C1	0.062	0.000 (4)	0.021 (1)
C2	0.204	0.000 (1)	0.031 (1)
C3	0.075	0.017 (2)	0.024 (1)

The PSO approach has the ability to find the optimum solution, but since it is based partially on random movement, it is possible to find a good solution but not the *best* solution. The all-terminal scenario presents interesting challenges because normally there are multiple unrelated vulnerabilities in the network that need to be mitigated.

Table 17: $V(N, E, r)$ - All Terminal (added nodes shown in parentheses), ILP Optimized shown for low and high weight options

Network	Initial	ILP Optimized	Swarm Optimized
A1	0.236	0.122, 0.000 (1,6)	0.070 (1)
A2	0.278	0.152, 0.000 (1,7)	0.141 (1)
A3	0.089	0.012, 0.000 (2,4)	0.019 (1)
B1	0.356	0.080, 0.017 (2,4)	0.082 (1)
B2	0.449	0.049, 0.004 (2,6)	0.049 (1)
B3	0.230	0.092, 0.000 (1,4)	0.047 (1)
C1	0.341	0.099, 0.0551 (3,7)	0.147 (1)
C2	0.329	0.133, 0.0146 (3,4)	0.091 (1)
C3	0.301	0.156, 0.0275 (3,11)	0.211 (1)

6.3.1 Network A1 - Point-to-Point Scenario

Figure 47 shows Network A1 (the point-to-point scenario). The source node is 2 and the destination node is 5. Figure 47(a) shows the geographic vulnerabilities in this network for the 2-5 pair. It is clear why the vulnerabilities are centered on nodes 10, 11&1, and 11&4. The vulnerability radius used is 2.

In Figure 47(b), the path formed by weighting the network to reduce the diversity cost to the right of the line from 2-5. The cost of a new node is $C = 1$ and $\alpha = 0.5$. Node 43 is added which eliminates the vulnerabilities caused by nodes 10, 11&4, and 11&1, but leaves a smaller vulnerability caused by nodes 11&6. Figure 47(c) shows the 4 initial swarm particle locations. The particle movement is shown. The vulnerability (0.028) was found and the movement stopped after a threshold of (0.03) was reached as shown in Figure 47(d).

6.3.2 Network B1 - All-Terminal Case

Figure 48 shows a 25 node example (Network B1) for the all terminal case. In Figure 48(a), we see the connectivity generated by the ITU-R Pedestrian Model with a random set of obstructions as shown. Each obstruction exterior wall contributes 6bB to a path. The non-optimized network in Figure 48(a) shows significant vulnerabilities in 4 areas of the network with a vulnerability radius of 2.

6.3.3 Network B1 - All-Terminal Case

Figure 48 shows a 25 node example (Network B1) for the all terminal case. In Figure 48(a), we see the connectivity generated by the ITU-R Pedestrian Model with a random set of obstructions as shown. Each obstruction exterior wall contributes 6bB to a path. The non-optimized network in Figure 48(a) shows significant vulnerabilities in 4 areas of the network with a vulnerability radius of 2.

6.3.4 Network C1 - All-Terminal Case

Figure 49 shows a 50 node example (Network C1) for the all-terminal case. In Figure 49(a), we see the connectivity generated by the ITU-R Pedestrian Model with random obstructions. The non-optimized network in Figure 49(a) shows significant vulnerabilities in the network and $V(N, E, r)$ of 0.341. A vulnerability radius of 1.5 was used.

Figure 49(b) shows the results of the ILP (spanning tree) approach with edge weighting and vulnerability of 0.099. The results of the PSO approach are shown in

Figure 49(c). The vulnerability by adding node 51 is 0.147.

6.4 Summary

The goal of this work was to develop methods to identify and mitigate geographic vulnerabilities in networks by adding nodes at locations that would minimize the vulnerabilities. Two scenarios were considered. The point-to-point scenario looks at providing geographic diversity between two paths on a given demand. The all-terminal scenario seeks to develop network configurations to protect the connectivity of all nodes in the network from geographic impacts. Two approaches were used. The first is an ILP approach that used a geographic weighting scheme to encourage path 1 to locate in a different geographic area of the network than path 2 and spanning tree 1 (all-terminal) from spanning tree 2. The second was a PSO approach that incorporated $V(N, E, r)$ as the *non-linear* objective. This was used for both the point-to-point and all-terminal scenarios.

Since $V(N, E, r)$ is non-linear, the ILP approach cannot use that function directly. Thus the ILP method produces results that approximate minimizing $V(N, E, r)$. The weight parameters in the ILP approach are strongly topology dependent. A small increase in the diversity weight can cause several nodes to be added drastically reducing $V(n, e, r)$. More research needs to be done to study the effect that the weight parameters have on solutions with different topologies.

The PSO approach adds one node at a time as it attempts to minimize $V(N, E, r)$ directly. This can produce optimal solutions. However, PSO is a heuristic and therefore it is possible to not locate the optimal solution. In our problem, generally we are looking

for *good* solutions not necessarily *optimum* solutions to produce low or zero geographic vulnerability. Given that assumption, the PSO approach seems to produce good solutions in most cases.

The disadvantage of the PSO approach over the ILP approach is the complexity of the PSO approach. In this work, we solved all problems using Matlab [96] with lp_solve [19] for for the ILP installed on a Windows XP desktop computer. $V(N, E, r)$ in the worst case calculates a shortest path calculation (point-to-point) or Eigenvalue decomposition (all-terminal), a maximum of $2N$ times, where N is the number of nodes. We are currently working on methods to reduce the complexity of the geographic evaluation function. In this work, the ILP approach calculation averaged approximately 1.0 seconds for the 15, 25, and 50 node problems. The PSO approach averaged 25.0 seconds for the 15 node problems, 172.0 seconds for the 25 node problems and 1500 seconds for the 50 node problems. The ILP approach has clear advantages in this regard.

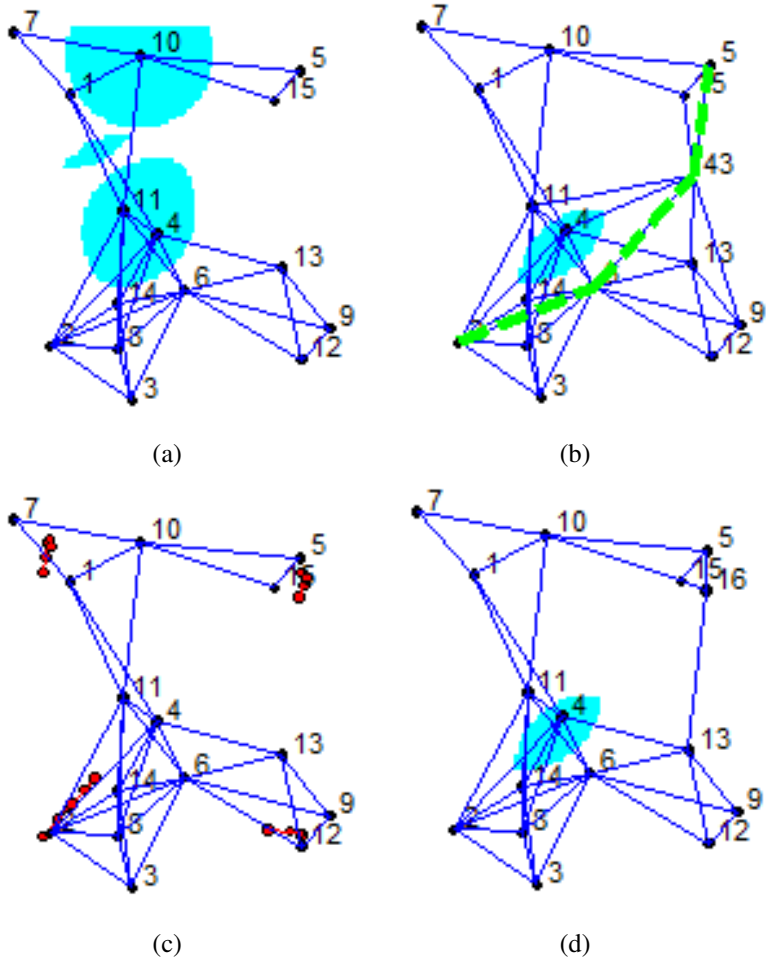


Figure 47: Network A1 Point-to-Point Example. (a) Network A1 Showing the V(N,E,r) of 0.182 prior to Augmentation. (b) ILP Optimized Showing Path 1 (Added Node 43), V(N,E,r) of 0.028. (c) PSO movement. (d) PSO Optimized Network (Added Node 16), V(N,E,r) of 0.028.

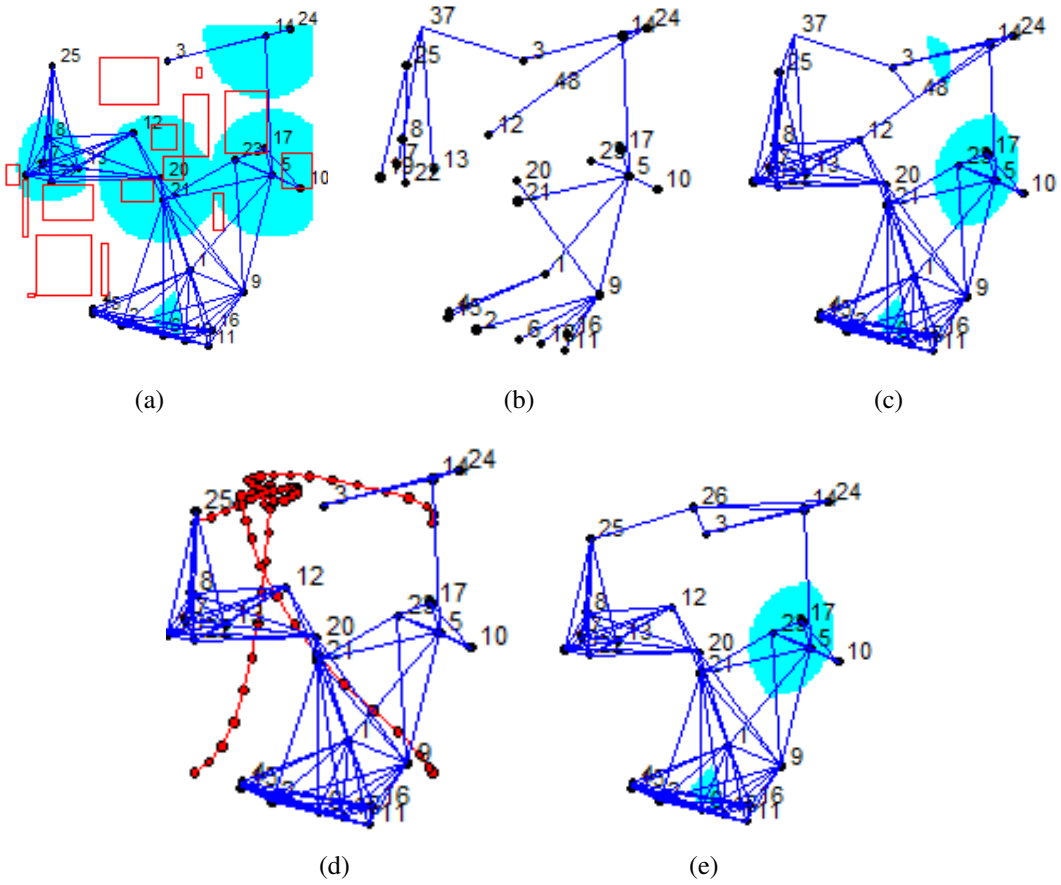
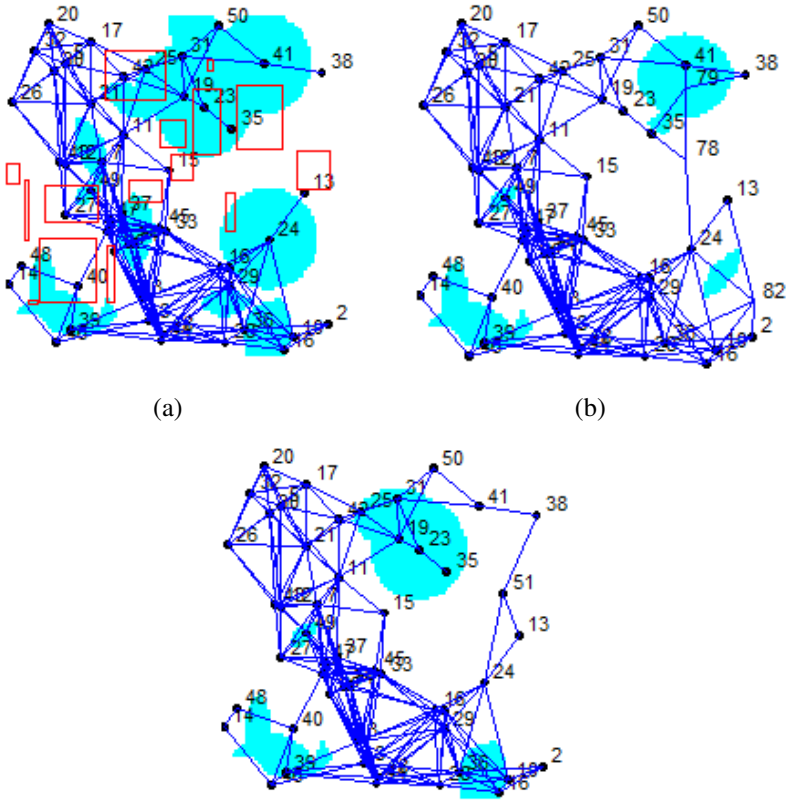


Figure 48: All-Terminal Example with ITU-R Model and Obstructions - Network B1 (25 Node) (a) Network B1 Showing the $V(N, E, r)$ prior to Augmentation (0.356). (b) ILP Optimized Outside Weighted Spanning Tree (Added Node 37, 48). (c) ILP Optimized, $V(N, E, r)$ of 0.090 (d) PSO Optimized (Added Node 26) (e) PSO Optimized, $V(N, E, r)$ of 0.082



(a) (b)

(c)

Figure 49: All-Terminal Example with ITU-R Model and Obstructions - Network C1 (50 Node). (a) Network C1 Showing the $V(N, E, r)$ of 0.341 prior to Augmentation. (b) ILP Optimized, $V(N, E, r)$ of 0.099 (Added Nodes 78, 82, 79). (c) Swarm Optimized, $V(N, E, r)$ of 0.147 (Added Node 51).

CHAPTER 7

ROUTING OF MISSION CRITICAL SERVICES DURING DISASTERS

Large scale geographical events such as natural disasters, major weather events, and geo-political events can significantly disrupt network services. Thus, efforts to reduce or minimize disruptions is desirable since such events are also when disaster response data is needed and when users want to inform their friends and families about their whereabouts. Therefore, it is imperative to create robust network functionalities.

From a network perspective, when geographically-correlated events occur, several problems surface. These problems range from slow convergence of the routing protocols due to congestion on overloaded links caused from rerouting traffic affected by the event. Furthermore, such events cause stability issues during the transient period. An important issue is how basic link-state interior gateway protocols (IGP) such as OSPF and IS-IS, that are commonly deployed in large networks, are affected and what can be done to increase throughput during a major event. In [49] and [16], the authors explore the tradeoff between routing stability and convergence speed in link state protocols. Disruptive outages in networks tend to cause multiple link state advertisements (LSA) to be flooded across the routing area and the routers to perform multiple shortest path calculations and therefore have long convergence times. *Flapping* can occur if link information is propagated through the network too quickly. To prevent flapping, hold down timers are used, but they slow down the convergence time [49]. This highlights the tradeoff between convergence

speed and routing stability. In addition, major disruptions can cause large numbers of traffic flows to be interrupted or rerouted to other links causing congestion [16].

Another concern during geographical events is that the failures may not occur at one instance. The outages may *cascade*, moving through the geographic area with the event or subsequent outages are caused by side effects of the initial outage, like congestion. In [147], Sterbenz, et. al. discussed several examples of major events where cascading outages were an issue. Earthquakes can have aftershocks that cause cascading outages. Floods can cause cascading outages as floodwaters penetrate different geographic areas. Even political situations can escalate and spread across a geographic region. Frequently, routing protocols will attempt to reroute with little knowledge that the paths that are selected are also vulnerable [147] or that the paths go around the edges of the disaster area.

In this chapter, we propose a preemptive geographic multi-topology routing (gMTR) approach based on multi-topology routing (MTR) [124, 125] to improve network performance during a major geographically correlated event. This work is published in [60, 62]. Basic OSPF and IS-IS protocols have now been extended with MTR functionality. Thus, our gMTR-based approach is intended to improve network throughput compared to basic link-state protocols in order to isolate parts of the network affected by a large-scale geographic event. Using our mechanism, there is less disruption to existing traffic, and the routing convergence time delay due to the basic link-state protocol can be avoided. Specifically, we use gMTR to protect against the problems associated with geographically correlated failures. Briefly, in the gMTR framework, we create a series of topologies that

can be useful to mitigate potentially disruptive events in the network.

7.1 Routing During Large Failures

To understand why gMTR is needed for geographic failures in networks, an understanding of what happens during link or node failures in link-state routing protocols is necessary. First, detection of a link or node state change (like an outage) occurs. Next, routers affected by the change will create LSAs and send them to their neighbors. This information will be flooded across the routing area till all routers have the same link state database. If database changes occurred, each router independently calculates the shortest path tree to all nodes based on the new database.

If multiple outages occur, like a geographic event, several things can happen that negatively impact the above restoration process. First, where multiple links are involved, links may exhibit repeated and intermittent failures causing link flapping behavior [16]. This behavior typically causes frequent routing changes that lead to routing instability. If the outages are significant, multiple rounds of LSAs will be flooded across the network causing shortest path first (SPF) calculations to run repeatedly at each node. This is commonly referred to as SPF Throttling and can lead to longer convergence times. A significant impact of route instability is that traffic paths may also become unstable and impacted severely [16]. Long `RouterDeadInterval` settings in routers also impact convergence times during major outages.

In this work, we show through simulation that `RouterDeadInterval` directly

impacts the time required to re-establish traffic across a network during a large geographic failure. Given that the reduction of this timer setting increases routing instability, a method to switch to a *stable* routing tree quickly is desirable to avoid significant traffic disruption during these events. We show that this can be achieved using gMTR.

In our gMTR approach, we create multiple alternate topologies starting from the default topology where link weights are pre-determined in such a way that in the event of a geographic failure, the routing paths created move away from a vulnerable geographic region. In particular, we propose a method to compute the link weights so that the path selection attempts to avoid an affected region. While the MTR-based approach has been used for network resilience [34,35,85], our approach considers a geographic vulnerability as the driver for how to determine links weights that can be used in MTR. This paper extends our earlier conference paper [60] in a number of ways: 1) we present two different ways to generate alternate topologies, one based on a circular coverage and the other based on a hexagonal coverage; 2) we include additional analysis on assessing improvements based on our approach; 3) we also include simulation results using the OPNET [118] simulation tool to show gains with gMTR during the transient period.

The general effect is that once the geographic event is detected, the *trunk* of the shortest path tree (SPT) is rapidly moved away from the affected area in the topology. This *isolates* the affected region from the rest of the network and limits the impact of the disruption on the whole network.

The primary reason to extend the coverage model to include hexagonal coverage is related to the well known efficiency of the hexagon coverage pattern making them useful

for sensor coverage and cellular networks. Hexagon coverage patterns have two important properties. Hexagonal tiling patterns have 100% coverage of a physical space as opposed to circular patterns. The other important property is that all locations in hexagonal coverage patterns are nearest to the center of the covering hexagon.

7.2 Related Work

Several schemes to improve convergence times and/or improve route stability have been proposed through the years. Modification of OSPF (or IS-IS) timers is the primary way proposed to prevent link or route flapping behavior. `Hello` timers and `RouterDeadInterval` timers can be extended to dampen flapping behavior. In addition, the interval between successive SPF calculations (SPF hold down timer) can be adjusted to prevent repeated SPF calculations. Clearly, increasing these timers will delay convergence times. These methods are discussed in detail in a number of works including [16, 36, 49].

Improvement in the outage detection time has also been proposed to improve stability and convergence [49]. This is most productive if hardware methods of detecting outages can be used instead of strictly relying on reducing the `Hello` timer. False alarms can cause link flapping and other instability. Additional information may also be available that could improve geographic outage detection times, like weather information or political events.

Incremental SPF (iSPF) has been proposed as a way to reduce SPF processing at the routers [97]. The idea behind iSPF is to incrementally change the shortest path tree

when link or node changes occur near the *leaves* of the tree. Performance improvements due to iSPF are discussed in [16, 49]. The algorithms we propose can take advantage of iSPF efficiencies.

In [93], one of the important observations of the impact of Hurricane Sandy is that the major internet providers moved the majority of the traffic away from New York City during the event. The goal was not to eliminate Internet access for the New York City area, but to reduce the possible impact the hurricane was having on traffic that was originally *passing* through the New York City area. We propose gMTR to quickly move traffic from a vulnerable geographic region.

There have been several concepts to modify routing mechanisms to accommodate disaster planning. In [68], Hanson et al. proposed a method of using routing layers to isolate areas of a network. The idea is similar to our work, except that they do not propose using Multi-Topology Routing to accomplish it. Instead they utilize the ability to segment networks into areas using protocols (like OSPF) that can have multiple areas. Their methods require that a routing area be pre-configured for possible geographic events. We opine that this was not flexible enough for robust disaster planning. In [107], the authors use a novel hill-climbing algorithm to assign link weights such that the link utilization is maintained at a level so that in the event of a disaster, re-routing would not *overload* links causing congestion. While this is useful for network provisioning, it does not necessarily prevent the routing churn caused by large geographic events.

Several local IP fast re-routing methods have been proposed that are summarized in [141]. In general, the router that detects the outage reroutes locally. There are also

methods where the packet carries information about the network elements that are to be avoided. These include the *not-via* methods as described in [134] and others like Localized On-Demand Link State Routing in [132]. These methods are good for a small number of link or node failures, but the mechanisms in these methods either do not support large numbers of correlated node/link failures or the methods do not scale well enough to support large scale outages.

Multi-Topology Routing (MTR) was originally proposed to enhance traffic engineering in link state protocols, OSPF and IS-IS [124, 125]. MTR allows different classes of traffic to use different virtual topologies such that traffic that requires higher Quality of Services could use a lower latency path than other traffic. In OSPF, the distribution of additional link weight information for different topologies is accomplished using the LSA. The Type of Service (TOS) field in the LSA is redefined as the MT-ID field. This provides link weight information for each topology to all of the routers in the network. A default (non-modified weighting) topology is required to be maintained.

When an IP packet is to be routed, the Differentiated Services Code Point (DSCP) field is used to note which topology is to be used for this packet. The router can look up the topology ID for this packet, check the shortest path tree (SPT) for this topology, and forward the packet based on this SPT.

It was found that MTR could be used for other purposes such as fault tolerant routing besides for use in traffic engineering for different services' classes. Approaches for implementing MTR for fault tolerance have been described by several authors including Menth and Martin [108], Cicic [34], and Scheffel et al. [138], and Kvalbein et al. [86].

Alternate topologies are used in these works primarily to protect against a single link or node failure. In [34], Cicic defines a legal topology with the following three criteria. All restricted links are attached to an isolated node and non-isolated node. The links between two isolated nodes are isolated. And the non-isolated nodes have no isolated links. Using an algorithm to create the topologies from this criteria, they are able to minimize the number of topologies.

Scheffel et al. [138] took a optimization approach to creating topologies. They created a binary integer linear program (BILP) to create topologies that had the goal of being able to mitigate single link failures with a given number of topologies. The objective was to minimize path distance.

In both works, once a link outage occurs the detecting node is responsible for finding a topology where the failed link is isolated. The packet can then be rerouted around the failed node or link. We refer to this as *local* rerouting based on MTR.

7.3 The Geographic Multi-Topology Routing (gMTR) Approach

With the goal to reduce the impact that a geographic event can have on a network, our approach is different than previous works. The topologies created here are intended to avoid routing (other than local routing) in a specific geographic area. The topology creation algorithm increases link weights around a specific geographic location with a specific radius. This has the affect of pushing the SPT away from the vulnerable area and, thereby, only allowing leaves of the SPT to exist in the vulnerable area. The link weight calculation algorithm decreases linearly from 100 at the avoided location till the

radius boundary and then the weights are set to 1. Although we have tested other weight calculation algorithms (such as exponential decrease), we found that the linear weight with a boundary at the radius seems to perform well. The advantage of using high (but not infinite) weights is that the SPT tries to avoid the vulnerable area using it only if no other route is available. Multiple topologies are then created with different geographic locations.

Once topologies are created, several options exist to select a topology and transition to that topology.

- Individual routers may choose the topology based on the early link state information that arrives at that router. A set of criteria to govern the selection of a topology that is not the default topology is presented.

- A protocol could be created separately from the routing protocol that has the responsibility of detecting geographic events rapidly and notifying routers in the network of the topology change. This would allow routing timers to be maintained at traditional stable levels.

- Network administrators may preemptively select a topology based on situational information that exists. This could be a significant weather forecast, declining political situation, or other large scale events that are known prior to the impact to the network.

7.4 Topology Creation Algorithms: gcMTR, gtMTR

We propose two methods to create multiple topologies: the Geographic Coverage Multi-Topology Routing (gcMTR) approach and the Geographic Targeted Multi-Topology Routing approach (gtMTR). In gcMTR, N topologies are created, each that can *avoid* a geographic area in a network. If these topologies are distributed accordingly, they can provide coverage against most geographic events.

The gtMTR approach uses existing knowledge about specific events to build a topology that *targets* a specific geographic area. Targets could include natural disasters that have a likely location (like an earthquake fault line or area frequently impacted by hurricanes), geo-politically vulnerable areas, or other anticipated geographic impacts. This enables network planners to anticipate certain geographic impacts. The notations used are:

- $G(V, E)$: Network Graph with nodes V and edges E (links)
- N: number of topologies required in gcMTR or gtMTR
- V_i: center of vulnerable area i
- R: radius of any vulnerable area
- T_i: MTR Topology i
- D_{ij}: the distance between V_i for T_i and the closest point on link j.
- $MaxD_{ij}$: the maximum distance across network

- W_{ij}: the Link Weight on link j for T_i.

Algorithm 1 (gcMTR) creates a set of topologies that provide coverage across a network to protect against against a variety of geographic events where the size of the event is known but the location of the event is not known. Function $genCoveragePlan$ generates locations to avoid based on a coverage plan that can be configured for a specific network.

Algorithm 2 (gtMTR) utilizes knowledge about a particular type of an event where the size and location of the event is known. Function $genTargetList$ generates locations to avoid based on a custom list of predefined geographic event locations.

Algorithm 10 Create Coverage Topologies - gcMTR

```
for all i in N do
  V_i ← genCoveragePlan(i)
  for all j in K links do
    if D_ij < R then
      W_ij ← Norm(MaxD_ij − D_ij)
    else
      W_ij ← 1
    end if
  end for
end for
```

The results in the following sections show that the greatest gain for gMTR methods occurs when the vulnerable area is well matched to the actual geographic event both in size and location. Therefore, if the event center was in a non-covered area, it is likely that the topology chosen would not be a good match for the event. In Fig. 51, a grid pattern is used to lay out the circles. The circle packing in a square method of coverage is shown using the Cost266 network from the Survivable Network Design Library (SNDlib

Algorithm 11 Create Target Topologies - gtMTR

```
for all i in N do
  V_i ← genTargetList(i)
  for all j in K links do
    if D_ij < R then
      W_ij ← Norm(MaxD_ij − D_ij)
    else
      W_ij ← 1
    end if
  end for
end for
```

1.0) [119]. It is easily shown that coverage of circles in a square is $\pi/4$ (78.5%).

A more efficient method of coverage would be to use a hexagonal approach. It is well known that the highest coverage of circle packing is possible using a hexagonal approach with a coverage of $\pi/(2\sqrt{3})$ (90.7%). The coverage difference using this approach is shown in Fig. 52. It is noted that even with this coverage approach, an event near the intersection of the hexagons is still not necessarily a good match to a given topology vulnerable area. A possible solution would be to overlay another hexagonal pattern of topologies on the existing layout, shifting the centers to align with the intersection of the hexagons. The topology with the center closest to the event center would still be chosen.

To use gcMTR, knowledge of the size of the geographic event that is anticipated is needed to determine the radius R. Once that is known, a coverage pattern can be selected. The two coverage patterns that are presented here are square (circle packing) and hexagonal. Figures 51 and 52 show how these coverage options would look on a typical network. In the square method, the topology centers V_i are laid out in a grid pattern with the distance between topology centers $Dist = 2R$, where R is the radius of

the desired circle. To design the hexagonal pattern, the distance between centers is two times the perpendicular distance from a topology center to the center of a flat edge on the hexagon. The hexagon pattern is staggered such the next row is centered between the topology centers of the previous row forming equilateral triangles. Both of these patterns are shown in Fig. 50.

Once the topology centers and the radius are chosen, weights are assigned based on (7.1). It is a simple linear scaling of the weights across a distance equal to the maximum distance across the network space. The modified weights are then applied inside of the circle targeted by that topology with 1 being applied elsewhere.

The gtMTR algorithm is similar to gcMTR with the exception that the topology V_i location is determined by prior knowledge of a specific event rather than a coverage pattern. The concept is similar, choose a radius and location. Calculate weights based on (7.1). Multiple gtMTR topologies can be created.

$$W_{ij} = \begin{cases} \frac{100(MaxD_{ij} - D_{ij})}{MaxD_{ij}} & \text{if } D_{ij} < R \\ 1 & otherwise \end{cases} \tag{7.1}$$

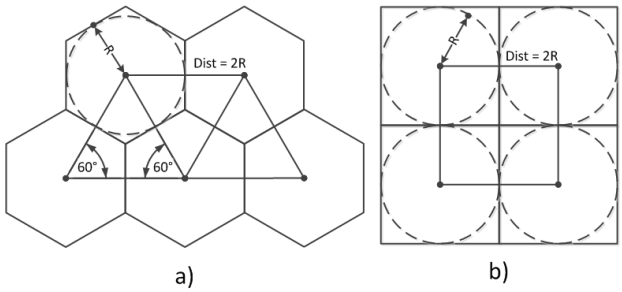


Figure 50: Coverage Pattern Design

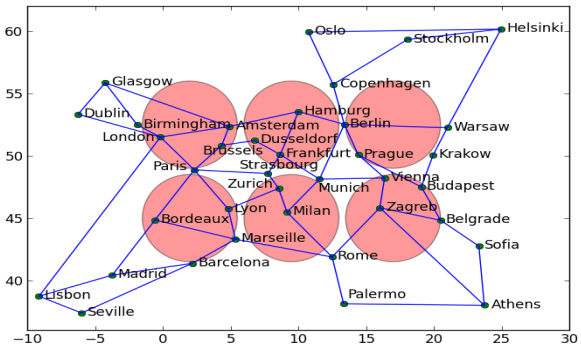


Figure 51: Topology Coverage Pattern (based on Circular Approach)

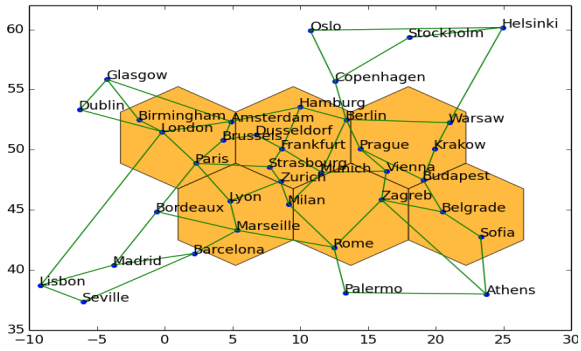


Figure 52: Topology Coverage Pattern (based on Hexagonal Approach)

7.5 Topology Selection and Use

As mentioned in Section 7.3, there are multiple operational concepts related to the selection of a topology. The Select Topology algorithm is relatively simple in its operation. Once a geographically correlated event is detected and location of that event estimated, the topology center that is closest to that location causes that topology to be selected. Details and discussion about defining and detecting geographically correlated events follow.

7.5.1 LSA Method

As OSPF (or other IGP) detects link and node failures via hardware methods and the `Hello` protocol, LSAs are generated and flooded across the routing area notifying the other routers of the link and node failures. Each router will use a set of criteria to enter the *MTR mode* and note a Geographically Correlated Event EV_i. The proposed criteria are:

1. More than 2 nodes or more than 3 non-adjacent links are out of service simultaneously within radius R. These devices form the basis of the geographic event with its center at the geographic mean of the failed nodes or links, and

2. The geographic center of the geographic event is within R distance of any V_i.

These criteria allow considerable flexibility to topology creation algorithms including overlapping coverage algorithms. If overlapping coverage is used, better fitting topologies should exist. The criteria also allow for a *no MTR mode* option that may be desired for certain geographic locations. It is assumed that all routers in the routing area are using the same algorithm (gcMTR or gtMTR). In addition, all routers would use the same topologies and the topology selection criteria. Routing loops would be possible only during the time between when the first router has changed topologies to the time when the last router changes topologies. One of the disadvantages of this approach is that `RouterDeadInterval` is still used for notification.

7.5.2 Geographic Detection Protocol

This approach overlays a protocol that monitors the network using other methods and when the criteria discussed in Section 7.5.1 are met, the protocol notifies the routers to switch to a different topology. The advantage of this approach is that faster switching could be accomplished since it is not as reliant on `RouterDeadInterval`, assuming other accurate and faster methods are used to detect router or link failures.

7.5.3 Situational Awareness

Perhaps the most promising use of our approach is when there is *awareness* of events in the different geographic areas of the network where our approach can aid in the decision making process to human operators. The advantage with this approach is that a preemptive switch to a different topology is possible at the appropriate instant, lessening the impact of the proposed event significantly.

Certain types of large geographic events can actually *spread*, causing routers near the edges to fail later than the routers near the center of the event. This could be the case with certain weather events like hurricanes or even political instability. In this case, the impact to routing in the network is more significant as route stability is further delayed. We tested this situation using simulation, as reported in Section 7.9.

Regardless of the operational concept chosen, to achieve *fast* switching during the event, routers must maintain SPTs for all topologies that could be used. In order to minimize processing at the routers, two approaches could be used. First, the SPT calculations of not-in-use topologies should be put in a lower priority than the in-use

topologies. Second, it may not be required that the not-in-use topologies be calculated as often as the in-use topology. If there were k topologies and we assume that t default topologies require SPT calculations before the not-in-use topologies SPTs are calculated, that would be k/t SPT additional calculations for every normal SPT calculation. Since, it is well known that Dijkstra shortest path algorithm has $O(n\log(n)+e)$ complexity, where n is the number of nodes and e is the number of edges (links), the overall complexity would increase linearly in the worst case with the number of topologies.

We now formally define a geographically correlated event:

DEFINITION 7.1. *$\langle EV_k, P_k, T_k\rangle$ is a Geographically Correlated Event meeting MTR mode criteria that is centered at point P_k with topology selected T_k.*

Algorithm 12 is proposed for the selection process.

Algorithm 12 Select Topology

```
if Geographically Correlated Outage Occurs, ⟨EV_k, P_k, 0⟩ then
  Vdist ← ∞
  T_k ← 0
  for all j in N Topologies do
    if D_jk < Vdist then
      Vdist ← D_jk
      T_k ← j
    end if
  end for
  ⟨E_k, P_k, 0⟩ ← ⟨E_k, P_k, T_k⟩
end if
```

7.6 Illustrative Example

To illustrate our approach, we now use the topology generation algorithms 1 and 2 on a 5×5 grid topology.

Fig. 53(a) shows the 5×5 grid topology with all link weights set to one, shown in green. Here, the SPT source node considered is 4. When geographic events occur, we assume that several nodes in an area are affected and disabled. This may affect the default SPT in a minimal way as Fig. 53(b) shows; only routing to nodes 20 and 21 was affected when nodes 10, 11, 15, 16 were disabled. In Fig. 53(c), the topology generated is shown when the vulnerability point is $V_i = (22, 57)$ with a radius of $R = 20$. Here, the topology moves the vulnerable point to the leaves of the tree.

However, when nodes 1, 2, 6, and 7 are disabled, Fig. 53(d) shows that routing to 10, 11, 12, 15, 16, 17, 20, 21, and 22 is affected by the SPT change. Clearly, the location of the vulnerability with respect to the main part of the default SPT affects the impact of the outage. Fig. 53(e) shows the 5×5 grid with a topology generated with a vulnerability point at $V_i = (40, 24)$ with a radius of $R = 20$.

7.7 Complexity of Approach

It is well known that Dijkstra shortest path algorithm has $O(n \log(n) + e)$ complexity, where n is the number of nodes and e is the number of edges (links). With this approach, the shortest path algorithm is executed more often before the event in order to maintain a complete set of SPTs at all nodes when the event occurs. If there are k topologies, k being constant, it will be executed $k(n \log(n) + e)$ times, while the overall

complexity remains at $O(n\log(n)+e)$.

Since these topologies are not used until the event occurs, we would recommend that the calculations of the additional SPTs be prioritized lower that the *in-use* topology (normally default) during SPT calculation. Furthermore, it is possible that the calculation of the SPT of the *not-in-use* topologies could even be delayed till the in-use topology has been updated a given number of times (t). It would reduce the execution time to $\frac{k}{t}(n\log(n)+e)$.

7.8 Analysis on Moderate and Large Topologies

To conduct further analysis of our gMTR algorithms, two networks were chosen. The first network is the Cost266 network, shown in Fig. 54, from the Survivable Network Design Library (SNDlib 1.0) [119]. It is a moderate size network topology with 37 nodes and 56 links. The second network is the AT&T Layer 1 topology generated by Sterbenz et al. [145] and is available at [84]. Fig. 55 shown this large physical layer topology with 383 nodes and 483 links.

Since one of the goals of the simulation is to determine the effect of the topology center difference from the event center, this analysis is applicable to both gtMTR as well as gcMTR. Assuming with gtMTR, one has knowledge of the location of the event, it should present results based on a small difference between the event center and topology center.

We computed and analyzed the following two metrics:

1. **Percentage of Dropped Connections** $(\%\mathbf{Drop})$**.** This is the percentage of all paths

(P_K) used to implement K demands that are interrupted immediately following the geographic event at $t = 0$ as shown in (7.2). This does not include demands disconnected because a node at either end of the demand was disabled by the event.

$$\%\mathbf{Drop} = \frac{P_K - P_{K(t=0_+)}}{P_K} \tag{7.2}$$

2. **Average path length** $(\mathbf{L}(\mathbf{P_K}))$**.** This is the average length of paths (P_K) used to implement K demands.

During simulation, we considered the time to re-establish services interrupted by the geographic event, referred to here as the convergence time. The gMTR *Gain* is defined as the convergence time $\times$ the bit rate interrupted with the default topology, giving us the effective amount of information not lost by using the selected topology as opposed to the default.

For the Cost266 network, SPTs were generated from all nodes as sources for the Dijkstra's algorithm and the results were averaged. The services are defined as connections from the sources to all other nodes. For the ATT L1 network, SPTs were generated from a selection of 30 nodes across the network as sources for the Dijkstra's algorithm. Thirty major metropolitan areas were chosen.

7.8.1 Cost266 Network

In the Cost266 network, a series of topologies with a circular coverage pattern and in a hexagonal coverage pattern was created to provide coverage plans for that network. Fig. 51 shows a circular pattern and Fig. 52 shows a hexagonal pattern of 6 locations (V_i)

used to create the coverage topologies. The closest (V_i) to the event center is the selected topology. Fig. 54(a) shows the Cost266 network Default Topology (all link weights $= 1$). For illustrative purposes, the SPT source in the figures is Lisbon. Three geographic events were created for evaluation as shown in Table 18. These were loosely based on possible disaster scenarios.

Fig. 54(b) shows the SPT with a Lisbon source after event 2. Amsterdam, Dusseldorf, and Brussels and all associated links were removed in this event. Fig. 54(c) shows the SPT with new link weights created by the selected topology using London as a source. The selected topology was 1 with a V_1 located at $(1, 51.0)$.

Table 19 shows the improvements gained by using a selected topology that was similar to the actual event. Generally, it was noted that if the event disconnects a major part of the default SPT, the improvements were significantly better than if the event was located in the *leaves* of the SPT. Event 2 disrupted the default path that crossed northern Europe. The circular topology that the was selected topology routed connections around this area and improved the number of disconnected services significantly. We found that the topologies generated by the hexagonal pattern performed better than the topologies generated by the circular pattern with the exception of Event 2.

Table 20 shows the average path distance prior to the event and after the event using the default topology, and after the event using the selected topology generated by the circular pattern or the hexagonal pattern. The distance was shortest during the default topology having more options than during the event or selected topology; however, the additional path length was not substantial.

Table 18: Cost266 Geographic Events

Event Number	X Location	Y Location	Radius
1	1.2	47.2	3.5
2	5.0	52.5	3.0
3	18.0	47.0	4.0

Table 19: Cost266 Evaluation - $\%Drop$ During Event

Event	Default Topology	Circular-based Topology	Hexagonal-based Topology
1	0.1093	0.0433	0.0000
2	0.2067	0.0164	0.1756
3	0.0703	0.0332	0.0000

7.8.2 ATT L1 Network

In the ATT L1 network, 8 topologies were created using the circular and hexagonal coverage plan for the network. Fig. 55(a) shows the circular pattern of 8 locations (V_i) and radii (R_i) used to create the coverage topologies. For brevity, we show details only for the topologies based on the circular pattern but the topologies used with the hexagonal pattern are shown in Fig. 56.

Fig. 55(b) shows the ATT L1 network Default Topology (all link weights $= 1$). The SPT source in the figures is New York. Four geographic events were created for evaluation as shown in Table 21. These were loosely based on possible disaster scenarios.

Fig. 55(c) shows the SPT with a New York source after event 1. Nodes and links in the New Madrid, MO, USA area were removed in this event, simulating a major earthquake along the New Madrid fault. In Fig. 55(d), the SPT created by the selected topology using New York as a source is shown with $V_i = (-92, 33)$, Radius $= 3.5$. Fig. 55(e) shows

Table 20: Cost266 Evaluation - Average Path Length (Hops)

Event	Pre-Event Default	Post-Event Default	Post-Event (Circular)	Post-Event (Hexagonal)
1	3.657	3.769	3.714	4.097
2	3.662	3.918	3.913	4.063
3	3.578	3.669	3.615	3.883

the SPT with a New York source after event 4. Nodes and links in the Chicago, IL, USA area were removed in this event. Fig. 55(f), the SPT created by the selected topology using New York as a source, is shown with $V_i = (-92, 41)$, Radius $= 3.5$.

In Table 22, we see that the $\%Drop$ actually *increased* using the selected topology based on the circular pattern for event 1. This occurred for two reasons. First, the topology selected was not matched well with the area of the event. The topology selected (see Fig. 55(a) location (-92,33)) was centered significantly south of the center of the actual event. Second, the event itself did not cut either major branch of the default SPT that traversed the network from east to west. The topology actually re-routed connections north into the area where the event occurred. This can be observed on Fig. 55(c) and Fig. 55(d).

Furthermore, in Table 22, it is noted that when event 4 did sever a main branch of the SPT, significant improvements were gained by switching to the selected topology during that event. The $\%Drop$ changed from 20.5% to 1.2%. The other two events showed significant improvements also. During the evaluations on the larger network, it appears that the selected topology tended to route traffic *further* around the vulnerability point than occurred during the event. This is confirmed in Table 23, the path lengths were

slightly longer during the selected topologies.

As with the Cost266 network, the topologies generated by the hexagonal pattern performed favorably compared to topologies generated by the circular pattern except for event 2. This is likely due to the improved coverage and better locations of topologies in the hexagonal pattern.

Table 21: ATT L1 Geographic Events

Event Number	X Location	Y Location	Radius
1	-89.5	36.6	3.8
2	-78.6	34.0	4.0
3	-119.7	35.6	4.0
4	-90.0	40.0	3.5

Table 22: ATT L1 Evaluation - $\%Drop$ During Event

Event	Default Topology	Circular-based Topology	Hexagonal-based Topology
1	0.2294	0.2700	0.0183
2	0.1266	0.0028	0.0086
3	0.0426	0.0279	0.0115
4	0.2053	0.0126	0.0058

Table 23: ATT L1 Evaluation - Average Path Length (Hops)

Event	Pre-Event Default	Post-Event Default	Post-Event (Circular)	Post-Event (Hexagonal)
1	14.378	14.956	15.272	16.302
2	14.162	14.351	14.556	18.186
3	13.022	12.898	13.082	13.252
3	14.391	14.743	15.878	15.133

7.8.3 Sensitivity to Event Size, Node Density, & Location

One of the areas of interest is how sensitive these methods are to events that are not located directly on a topology from a location or size perspective and how sensitive these methods are to the density of nodes in a topology. The first test to evaluate this was to change the location of all 4 events in the ATT network to locations west and east relative to the center of the topology by 50%, 100%, and 200% of the selected topology radius. Fig. 57 shows the $Drop\%$ using both the default topology and the selected topology when the event center is not aligned with the topology center. The results were as one would expect. The selected topology performed well when the event was centered where the topology is centered. As the event drifts away from the center of the topology, the performance degrades approaching the performance of the default topology.

The second test varied the topology size of all 4 events in the ATT network from 25% to 200% of the event radius. Fig. 58 shows the average percentage of connections dropped using both the default topology and selected topology when the event size is not aligned with the topology size. When the topology size is between 100% and 200% of the event size, gMTR performs well. Performance degrades quickly outside these bounds. We also looked at the performance when the average number of nodes in a topology is varied. Fig. 59 shows the average number of nodes per topology versus $Drop\%$. This was done by reducing the topology size and event size concurrently. Fig. 58 and Fig. 59 are very similar. This leads to the following important observations. First, there is a definitive practical lower bound on the number of nodes per topology. It appears to be near 4. Second, the upper bound is related to the portion of the entire network that is covered by a

single topology. As the topology covered more than 70 out of 383 nodes (approximately 18%), the performance approached default performance. That would imply that a single topology should not cover more than approximately 15% to 20% of an entire network.

7.9 Simulation Study

To better understand the convergence process used by OSPF and gains that are possible using gMTR, we conducted a simulation study on the larger topology, the ATT L1 network. The goal of the simulation was to see how quickly traffic, which was interrupted by a geographic event was restored using the OSPF routing protocol in a large network and the gain obtained due to gMTR. During simulation, we considered at the time to re-establish services ("convergence time") interrupted by the geographic event. The gain using our gMTR approach can be simply identified as the product of the convergence time and the bit rate interrupted with the default topology versus the gMTR-selected topology. The simulation was built using OPNET [118]. All 383 nodes were built using a Cisco 12000 series router as the model. All of the ports were set up to use OSPF routing. The network was connected as per the ATT topology using 10 Mbps point-to-point Ethernet connections.

We constructed two scenarios. The first one was a single pair of demand between Pittsburgh, PA and Salt Lake City, UT. The traffic for this demand was a constant bit rate (UDP) approximately at 32 Kbps. The second scenario was a full mesh of demands between 12 cities. These are listed in Table 24. In both scenarios, a large geographic outage was scheduled to occur at approximately 200 sec into the simulation. The outage

is the same as is depicted in Fig. 55(e).

In the simulation, the gMTR topology that was used is topology 3 as shown in Fig. 55(d). This is the selected topology using Algorithm 12. Link weights in the selected topology were determined by our algorithm.

In all cases, the traffic was not initiated until approximately 100 sec into the simulation. This is to allow the routing protocol to initially converge prior to data transmission. The traffic noted at approximately 50 sec is routing protocol traffic. In all cases, except where noted, the `HelloInterval` was set to the default of 10 sec and `RouterDeadInterval` was set to 40 sec.

Table 24: ATT L1 Simulation - Multiple Demand Cities

New York, NY	Washington, DC
Atlanta, GA	Miami, FL
Chicago, IL	Dallas, TX
Minneapolis, MN	Pittsburgh, PA
Salt Lake City, UT	Seattle, WA
San Francisco, CA	Los Angeles, CA

7.9.1 Single Demand Simulation

Figs. 60 - 63 are related to the default topology and how gMTR would work with an outage occurring in the region surrounding Chicago, IL, which was depicted in Fig. 55(e). In Fig. 60 we see the traffic path used by the default topology prior to the event at 200 sec. This path is through Chicago. The traffic path used by the default topology after the event occurs is also shown, going though Kansas City, MO. In Fig. 62, the throughput is shown for the demand. It is easy to see the 40 sec interruption in traffic

when the outage occurs.

Fig. 61 shows the path used due to the gMTR-selected topology. The path takes a wider route around the topology 3. Finally, as expected, Fig. 61 shows the un-interrupted traffic flow through the outage.

To further explore the influence of `RouterDeadInterval` on the convergence time, we conducted additional tests. This tests were run with a `HelloInterval` of 2 sec and `RouterDeadInterval` of 10 sec. As shown in Fig. 64, the convergence time was reduced from 40 sec to approximately 10 sec, which follows the `RouterDeadInterval` settings. This confirms the relationship between the settings for `RouterDeadInterval` and the convergence time in OSPF.

It raises the question of why one would not just reduce `RouterDeadInterval` in OSPF to achieve faster convergence times? As was pointed out in Section 7, reduction of the OSPF `RouterDeadInterval` timer reduces the stability of OSPF and can cause *flapping*. The methods described here reduce convergence and are found to be stable.

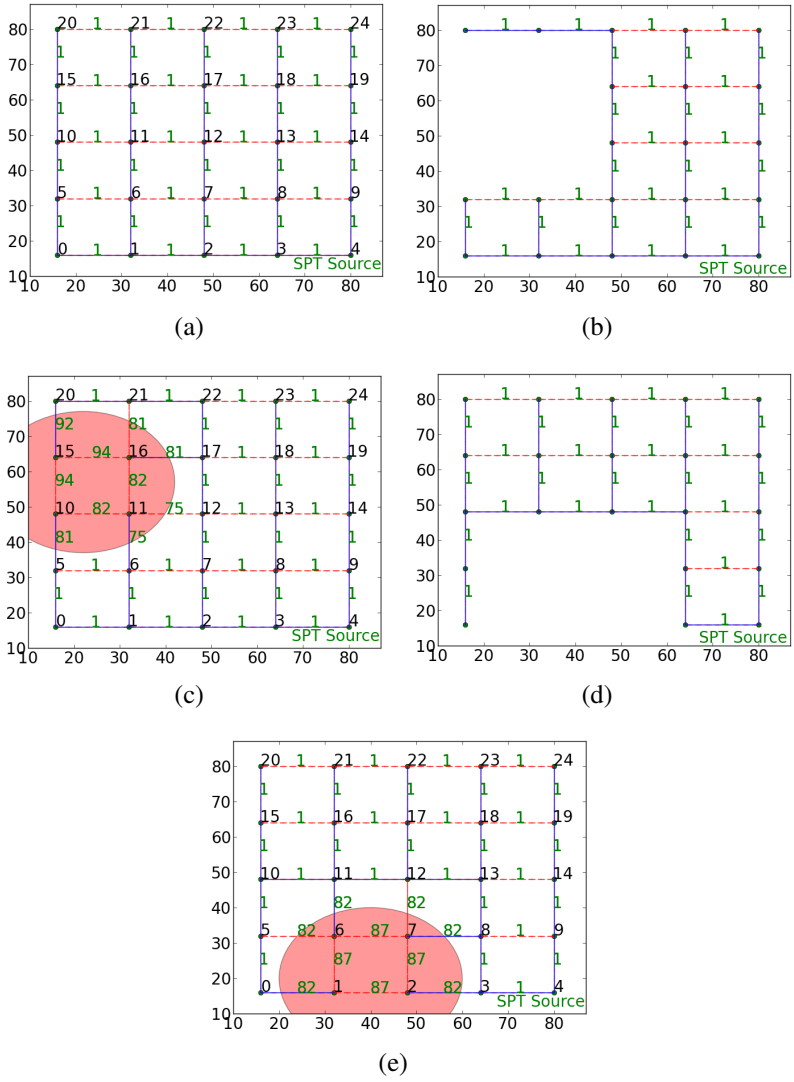
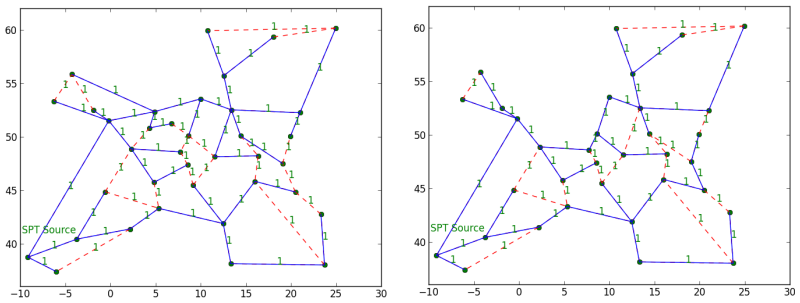
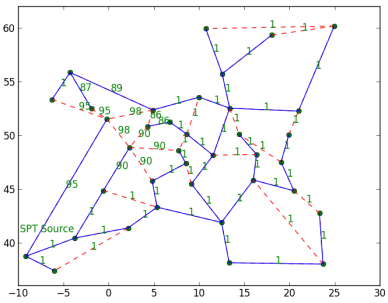


Figure 53: Grid5 Network, SPT Source $= 4$, SPT links are solid, non-SPT links are dashed. Link weights are shown. (a) Grid5 Default Topology (b) Network with nodes surrounding $(22, 57)$ with radius $R = 20$ deleted (c) Topology created with $V_i = (22, 57)$ with radius $R = 20$ (d) Network with nodes surrounding $(40, 24)$ with radius $R = 20$ deleted (e) Topology created with $V_i = (40, 24)$ with radius $R = 20$



(a) (b)



(c)

Figure 54: Cost266 Network, SPT Source = Lisbon, SPT links are solid, non-SPT links are dashed. Link weights are shown. (a) Default Topology, (b) Event Scenario 2, Location $= (5, 52.5)$ Radius $= 3.0$, (c) Selected Topology with $V_i = (1.0, 51.0)$, Radius $= 4.25$.

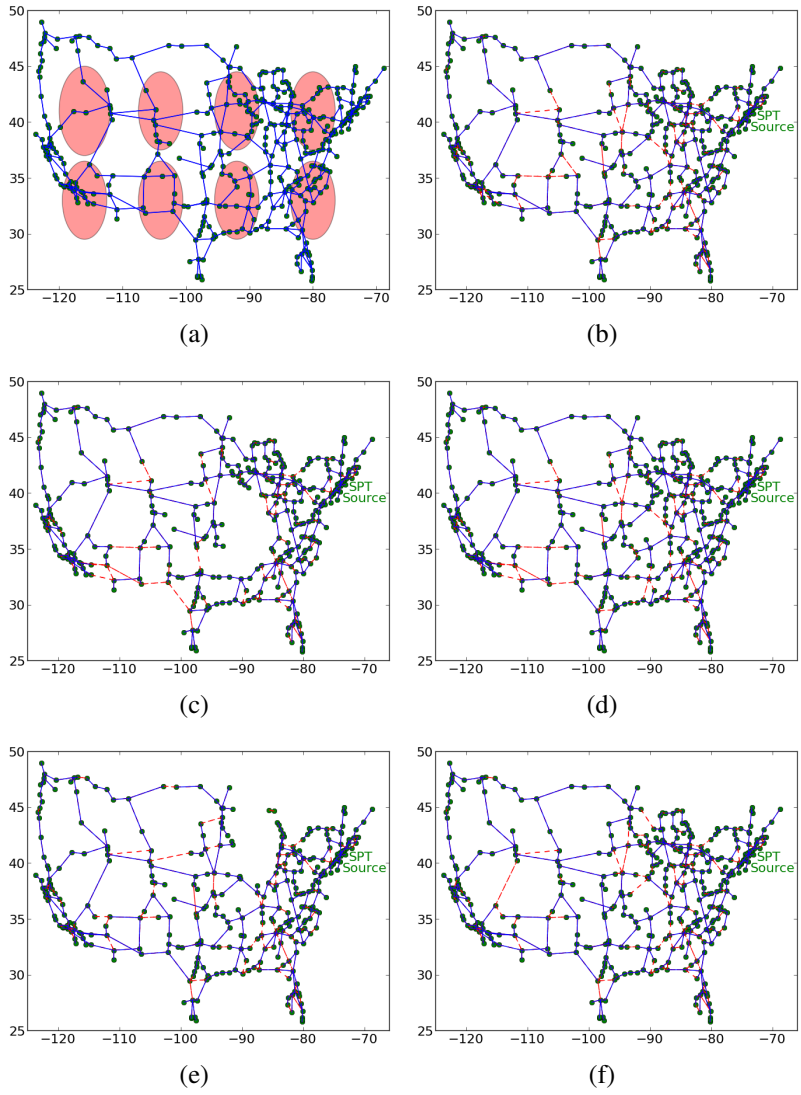


Figure 55: ATT L1 Network, SPT Source = New York, SPT links are solid, non-SPT links are dashed. Link weights are shown. (a) Topology Coverage Plan (b) Default Topology (c) Event Scenario 1, Location = $(-89.5, 36.6)$ Radius = 3.8. (d) Selected Topology with $V_i = (-92, 33)$, Radius = 3.5 (e) Event Scenario 4, Location = $(-90, 40)$ Radius = 3.5. (f) Selected Topology with $V_i = (-92, 41)$, Radius = 3.5

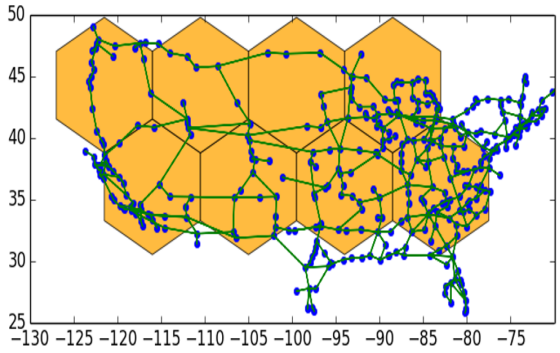


Figure 56: ATT L1 Network, Hex Topology Coverage Plan

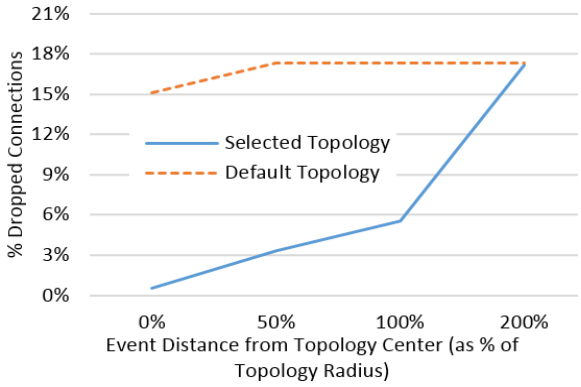


Figure 57: Event distance from Topology Center versus $\%Drop$

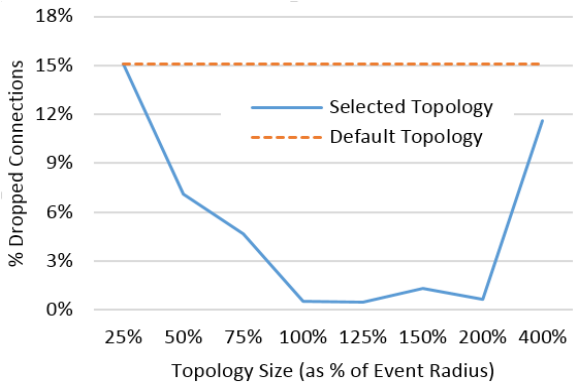


Figure 58: Topology Size compared to Event Radius versus $\%Drop$

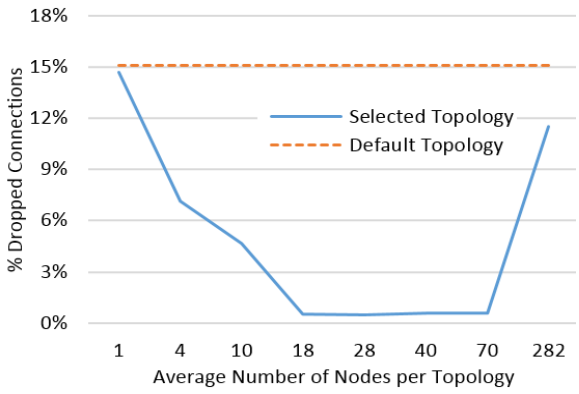


Figure 59: Average Nodes per Topology versus $\%Drop$

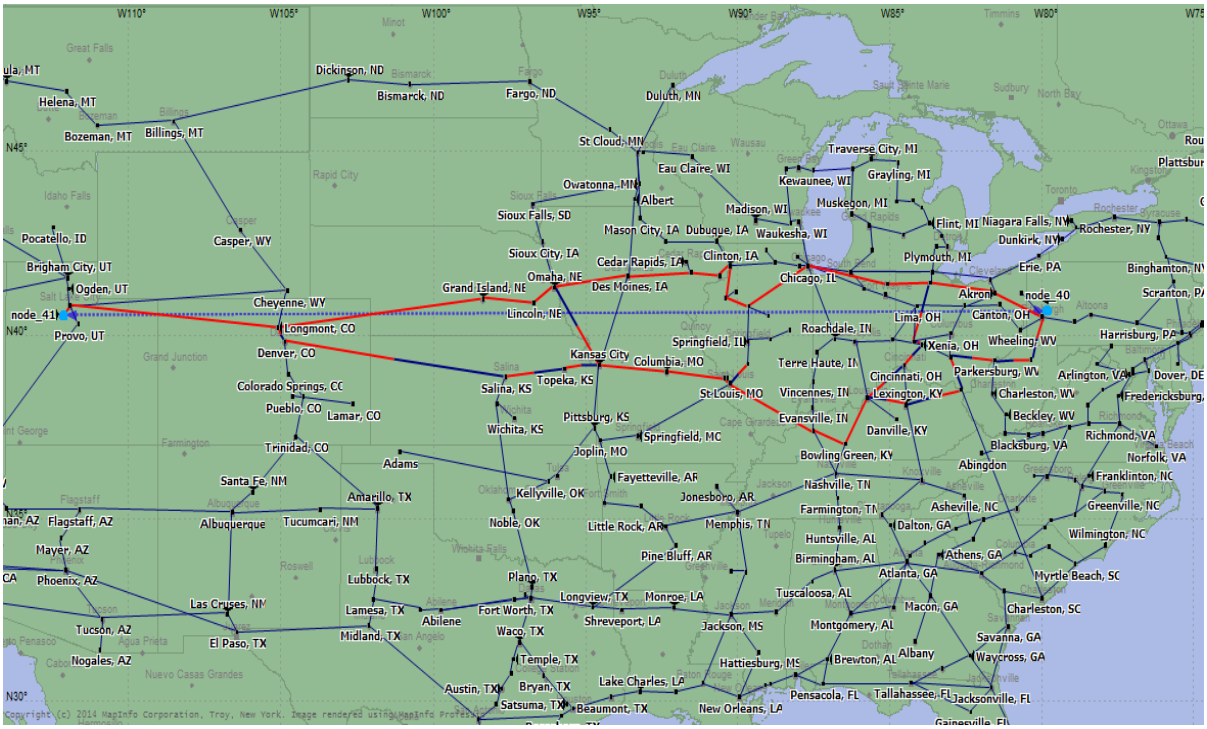


Figure 60: Single Demand Showing Pre-outage and Post-outage Paths

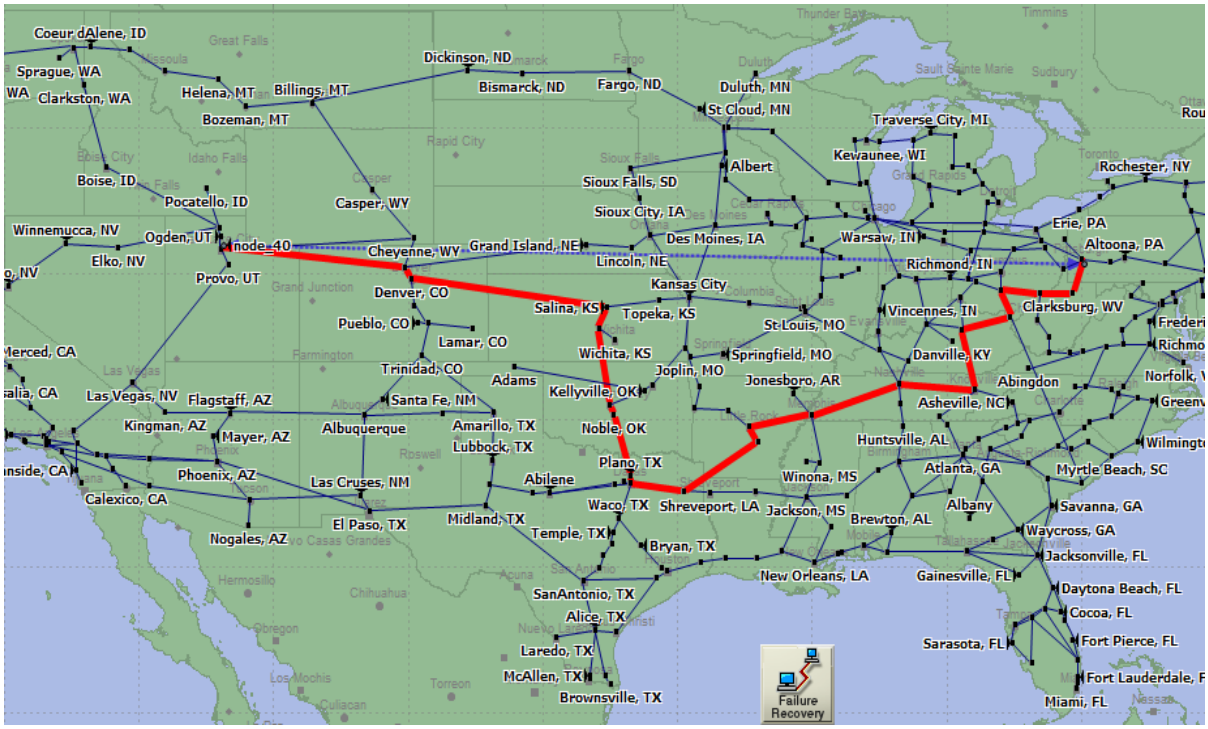


Figure 61: Single Demand Using Selected Topology Routing based on gMTR

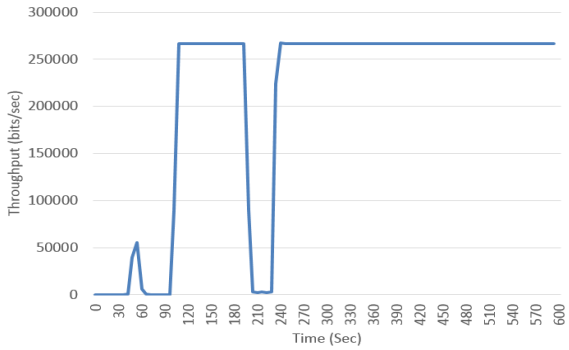


Figure 62: Throughput Using Default Topology (Single Demand)

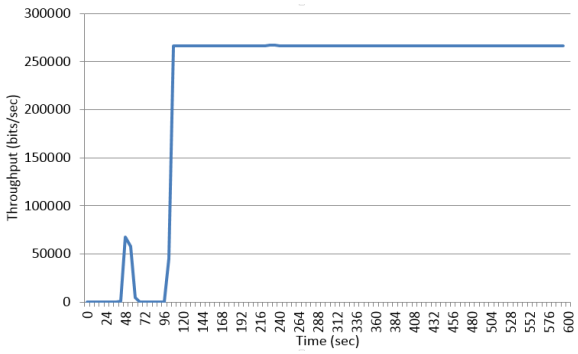


Figure 63: Throughput Using Selected Topology based on gMTR (Single Demand)

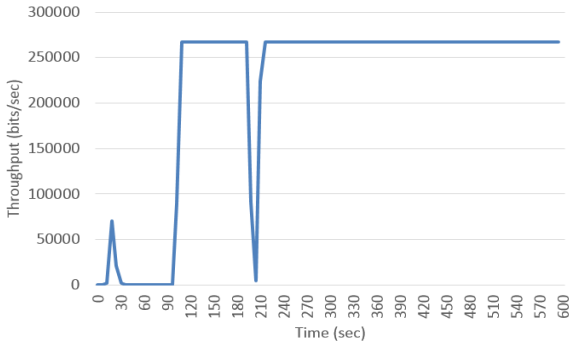


Figure 64: Throughput Using Default Topology and Modified OSPF Timer Settings (Single Demand)

7.9.2 Multiple Demand Simulation

Figs. 65 - 27(b) are related to how OSPF would function with a large geographic outage near Chicago, IL prior to and after gMTR is implemented as is depicted in Fig. 55(e). In Fig. 65 we see multiple traffic paths used by the default topology prior and after the event at 200 sec. Prior to the event, much of the traffic from Pittsburgh and New York that is destined to Seattle and San Francisco traverses through Chicago. After the event, the traffic shifts to a more southerly route through St. Louis.

Fig. 67 shows the received traffic at Salt Lake City from the other 11 sites using the default topology during the event. It is worthwhile to note progression through the event. At 200 sec, the traffic that crosses Chicago is lost. At approximately 230 sec, the traffic that was rerouted around Chicago is restored. The final level is the the original traffic minus the traffic that was terminated at Chicago. Fig. 68 shows the throughput on the Bridgeton, MO (near St. Louis, MO) to Columbia, MO link. This is the link that would have acquired much of the rerouted Chicago traffic. The increase in traffic is noted at approximately 240 sec. Finally Fig. 69 shows the traffic demands between Salt Lake City and one of the following cities: Dallas, Chicago, or Pittsburgh. The demand traffic reacts as expected during this event.

With gMTR implemented, Figure 66 shows the differences in demand paths. It is worthwhile to note the wider path around the chosen gMTR topology that is used, steering the traffic as far south as Dallas. It is apparent that with gMTR, a margin of error is implemented in case the outage is not exactly as predicted.

Fig. 70 shows the received traffic at Salt Lake City post-gMTR implementation

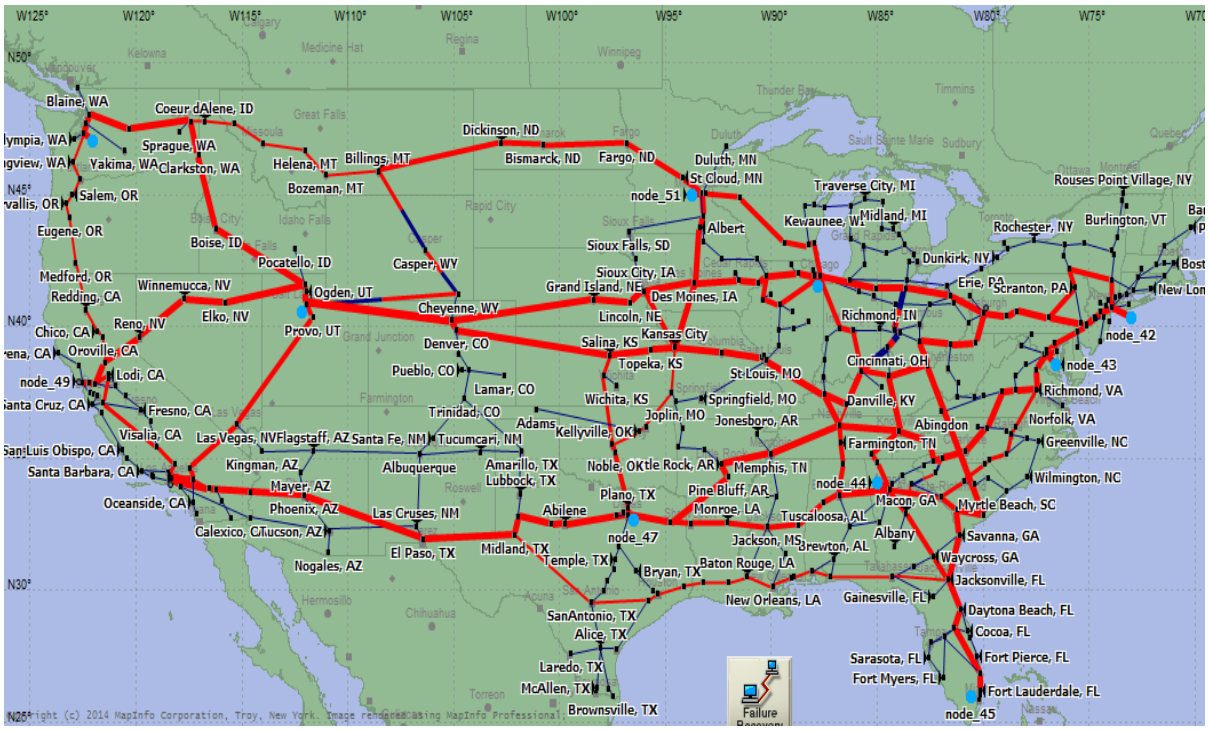


Figure 65: Pre-outage and Post-outage Paths Using Default Topology (Multiple Demands case)

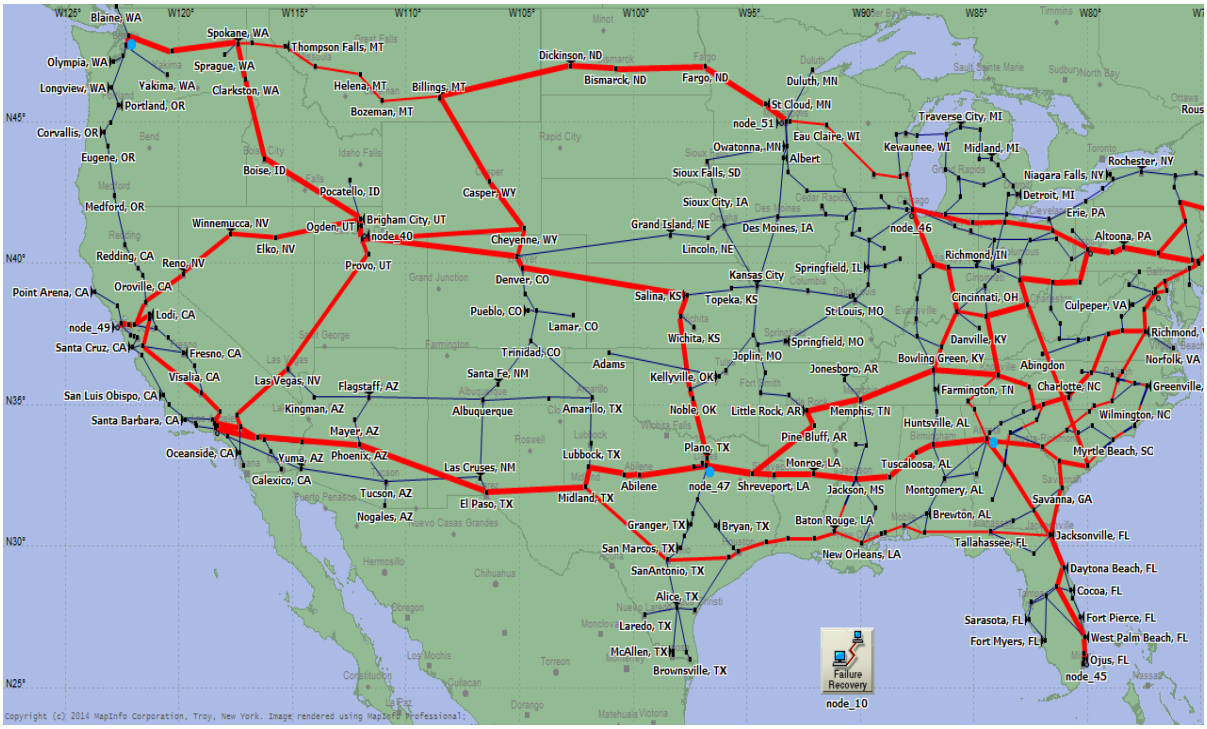


Figure 66: Pre-outage and Post-outage Paths Using Selected Topology based on gMTR (Multiple Demands case)

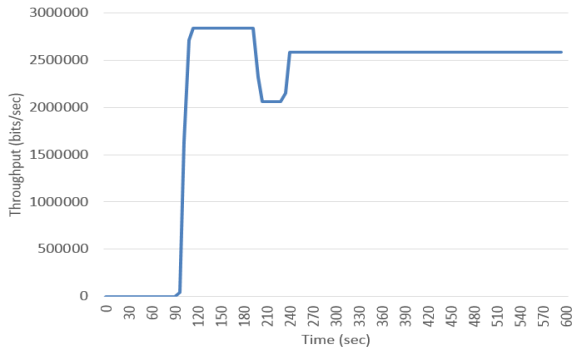


Figure 67: Receive Traffic at Salt Lake City, UT (Multiple Demands case)

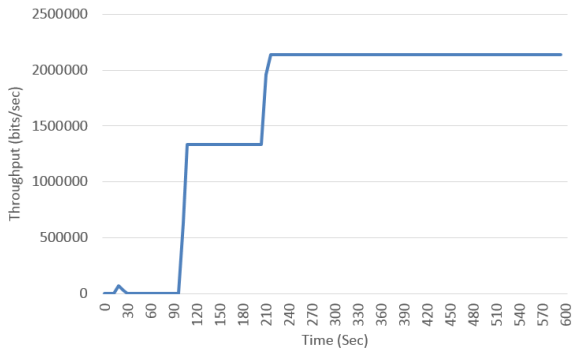


Figure 68: Throughput on Bridgeton, MO to Columbia, MO Link (Multiple Demands case)

during the event. When the event occurs, the only change that occurs is the loss of the traffic associated with the Chicago node. This illustrates the reduction of *churn* in the network when using gMTR.

Simulation results discussed so far considered outages as a single event. However, it is unlikely that in real life they would appear as a single event. Cascading outages that are geographically correlated have been documented and are discussed in Section 7. It is useful to investigate the harm caused by cascading outages. A cascading outage was

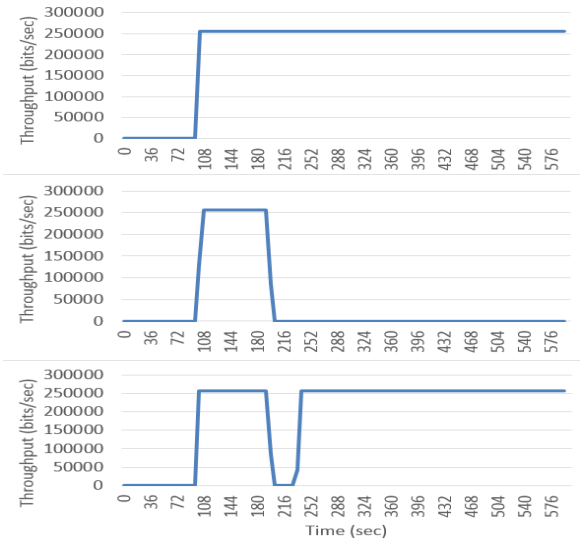


Figure 69: Traffic Using Default Topology between Salt Lake City, UT and a) Dallas, TX b) Chicago, IL c) Pittsburgh, PA (Multiple Demands case)

constructed by allowing the Chicago node to fail at 200 sec and the remaining nodes related to that geographic outage to fail at 260 sec. This caused the notable (but predictable) results between nodes at Minneapolis and Pittsburgh shown in Fig. 71. The traffic flow was interrupted initially at 200 sec, restored at approximately 240 sec, and failed again at 260 sec, and finally restored at approximately 300 sec. This illustrates the benefits of using gMTR for cascading events to provide a stable path for traffic that avoids the geographic area.

7.10 Summary

Through this work, we demonstrate that geographic multi-topology routing (gMTR) can be used to improve performance of routing during large geographic events like natural disasters. These improvements are achieved by maintaining additional topologies created

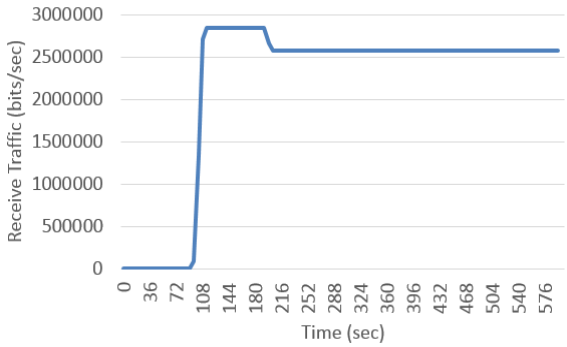


Figure 70: Receive Traffic at Salt Lake City, UT using Selected Topology based on gMTR (Multiple Demands case)

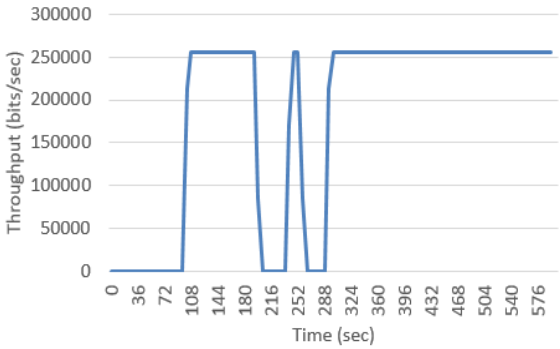


Figure 71: Demand Traffic Between Pittsburgh, PA and Minneapolis, MN during Cascading Failure

by increasing link weights in specific areas of the network through the gMTR approach. This has the effect of pushing the trunk of the Shortest Path Tree out of the region of highest impact, preventing disruption of much of the traffic during the event.

Multiple topologies can be created to anticipate multiple events in a network. When this is done, a topology is selected when it is apparent that a geographic event has occurred. The traffic is moved to the new SPT quickly and the impact of the event is minimized. Essentially, the amount of network *churn* is reduced during the geographic

event.

Clearly the key to improving routing performance during geographic events is to have knowledge of that event and be able to act on that event prior to the rerouting process used by OSPF, which is constrained by several timers intended to provide more stability in the routing algorithm. `RouterDeadInterval` and `HelloInterval` are key timers that can directly influence, not only the convergence time in the network but also the stability in the network. By using fast detection or forecasting techniques and switching to a known stable SPT, the convergence time can be minimized using OSPF.

We proposed three algorithms. Two algorithms gcMTR and gtMTR provided methods to create topologies in both a network wide coverage approach and targeted approach that can be used to anticipate a specific event, where knowledge of that event exists. gcMTR is implemented using a circular and a hexagonal approach. The third algorithm specifies a way to detect a geographic event and select a topology to use. A discussion is included of operational models that can be used to detect geographical events.

During evaluation, it was discovered that the number of dropped connections is typically significantly better if a selected topology is used. The exception to this is if the condition exist where (1) the event does not disrupt a significant branch of the default SPT *and* (2) the selected topology does not fit well with the event from a location perspective. We explored this further by modifying the distance that the event occurs from the center of the topology and by modifying the size of the event compared to the radius of the topology. Although not unexpected, the results emphasize the importance of the algorithm and parameter selection. An important observation is that there appears

to be practical bounds related to topology size and the distribution. If a topology does not contain more than approximately 4 nodes, the performance degrades to default and if a single topology covers a significantly portion of the overall network, the performance also suffers. These are both intuitive observations, and our testing provides a guide during implementation of topologies. Topologies generated by the hexagonal pattern tended to outperform topologies generated by the circular pattern in the networks tested here. The better coverage and more optimal locations of topology centers is likely the reason for this advantage. More research in the distribution of topologies is needed to determine the optimum configuration.

We used simulation to investigate the effects of geographic outages in networks using OSPF without gMTR and with gMTR. We analyzed the convergence times, defined as the time to restore traffic flows after the event, using default settings for `HelloInterval` and `RouterDeadInterval` timers in OSPF. The settings are then modified to understand the relationship between the timers and convergence times. Finally, a cascading geographic failure was simulated to show the impact of these types of failures.

CHAPTER 8

CONCLUSIONS AND FUTURE RESEARCH

In this research we looked at methods of providing resiliency against geographic vulnerabilities in mission critical networks. As we have seen, this is a multiple step process involving many components. In this this work, we looked at many of these steps. The FAA SWIM network was used as a real life example of a mission critical network involving air traffic control. Before discussing the individual steps in the resilience design process, we note that a key opportunity for future research is the assimilation of several of these methods into a singular set of tools. For example, the NIR involves the selection of a network test. The LAQT performance method discussed in Chapter 3 would be an interesting network test. Another interesting combination would be to incorporate the provisioning and restoral methods referenced in Chapter 5 with the NIR which would enable the NIR to capture not only the vulnerabilities but also the ability of the network to restore the most critical services.

First, we investigated the QoS in a mission critical multi-layer network. The two methods that we created are both based on network decomposition. The first method is an extension of QNA. The second method was a network decomposition approach based on Linear Algebraic Queueing Theory (LAQT). We were able to show that both methods can be used to approximate multi-layer networks. The LAQT method is relatively more accurate than the QNA method. Errors between simulation and the analytic methods

are suspected to be related to correlation between the arrival and departure process and correlation in the departure process. Future research directions include capturing various types of correlation in the LAQT model in a tractable manner.

Geographic vulnerabilities in networks were evaluated next. The NIR metric and SP-NSG algorithm were created to help evaluate networks for geographic vulnerabilities. During the SP-NSG process, one of the important benefits of state space pruning is that it can be used to find geographic vulnerabilities in networks. This benefit is exploited in this work. The NIR has the primary benefit that it can be used with different network tests, which allows the NIR to be configured specifically to the mission of the network and its applications.

However, for large networks, the vulnerability analysis can still become intractable. To maintain tractability in large networks, we also present a K-means clustering method. It relies on the idea of reducing the number of nodes for purposes of the state based analysis using clustering. Matrix transformation is used both for the K-Means Clustering Approach and the multilayer testing method that maps geographic failures simultaneously onto multiple layers. There are several areas of future research that can be explored related to the NIR and SP-NSG. A primary extension of this research would be the use of additional network tests like the performance methods described in Chapter 3. More research related to model selection and specifically the selection of K is also needed and discussed in Section 4.5.1.

Provisioning and restoration methods during disasters were investigated next. We propose a framework to use Service Level Agreements (SLAs) to provide much of the

information necessary to solve the challenging problems of remediation during geographically correlated events like natural disasters. We form the solutions to these problems as provisioning and rerouting solutions to limit outages on the most critical services due to geographic events. From the SLA parameters: system response time, availability, and survivability, we can create integer linear programs and heuristics to solve the provisioning and rerouting problems.

The MILP and heuristic approaches to the network provisioning problem with geographic challenges are both found to be effective solutions. The performance of both approaches are similar. The cost of the MILP approach varies from slightly less expensive to moderately less expensive than the heuristic solution. The MILP and heuristic approaches to the rerouting problem have significant differences. The heuristic approach performs slightly better than the MILP solution for the emergency services. But, the normal services see significant benefit under the MILP rerouting solution. The MILP formulation provides a better overall solution to the rerouting problem. This research could be extended significantly in a couple of ways. First, it would be interesting to explore the effect that topology selection has on the performance of the algorithms. Second, it would be useful to extend these algorithms to include multi-layer networks.

The goal of the topology improvements discussed in Chapter 6 was to develop methods to identify and mitigate geographic vulnerabilities in networks by adding nodes at locations that would minimize the vulnerabilities. Two scenarios were considered. The point-to-point scenario looks at providing geographic diversity between two paths for a given demand. The all-terminal scenario seeks to develop network configurations

to protect the connectivity of all nodes in the network from geographic impacts. Two approaches were used. The first is an ILP approach that used a geographic weighting scheme to encourage path 1 to locate in a different geographic area of the network than path 2 and spanning tree 1 (all-terminal) from spanning tree 2. The second was a PSO approach that incorporated $V(N, E, r)$ as the *non-linear* objective. This was used for both the point-to-point and all-terminal scenarios.

There were benefits to both approaches. The ILP approach is significantly faster but only estimated $V(N, E, r)$. The PSO methods used the $V(N, E, r)$ directly, but was expensive from a processing approach. In addition to testing a more extensive selection of PSO parameters (like the number of starting solutions), we would like to test a larger diversity of topologies including topologies that are better suited for wired networks. Also, these methods could be extended to include new links without new nodes (especially in the case of wired networks).

Finally, with respect to routing during geographic failures, we were able to demonstrate that geographic multi-topology routing (gMTR) can be used to improve performance of routing during large geographic events like natural disasters. These improvements are achieved by maintaining additional topologies created by increasing link weights in specific areas of the network through the gMTR approach. This has the effect of pushing the trunk of the Shortest Path Tree out of the region of highest impact, preventing disruption of much of the traffic during the event. Finally, optimization of the routing topologies (size, shape, and location) and frequency of calculating shortest path trees for not-in-use topologies could be interesting extensions of this work.

APPENDIX A

PUB/SUB SIMULATION MODEL DESCRIPTION

A.1 Message Routing/Forwarding Node

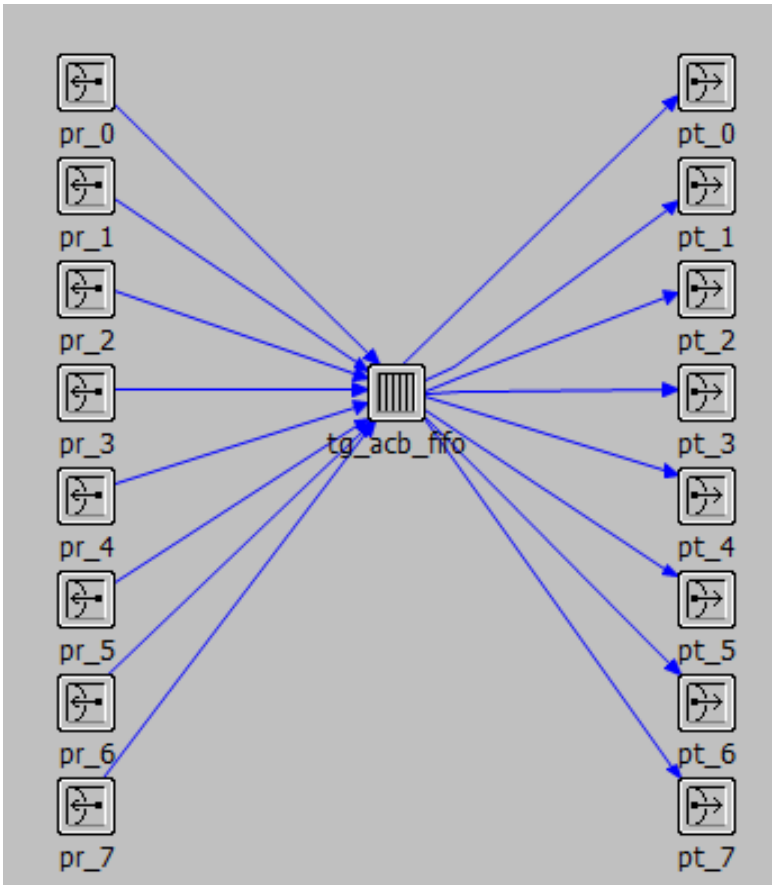


Figure 72: Routing/Forwarding Node

The message routing and forwarding node is shown in Figure 72. It includes standard Modeler transmit and receive interfaces. The process node is shown in Figure 73. It is a standard Modeler FIFO queue with modifications to copy the messages to send

out on multiple destination nodes. A more practical implementation may include a set of input and output queues instead of a single FIFO queue. But, this is a better model to validate and verify the analytical model.

Pub/sub message routing and forwarding is achieved via a CSV text file that contains the routing information for each product at each node. Table 25 shows the contents of a portion of the routing file. A message from product 1 arrives at node 1 from source 1 as shown in Figure 13. From there the product is copied to destination ports 2,3,4 which happen to go to nodes 3,2,4 as is shown on the first 3 lines of Table 25. Line 9 shows that a message from source 2 entering node 2 would be routed to node 6. It is easy to see how the one-to-many feature of pub/sub is implemented.

Table 25: Scenario 1-3 Configuration

Node	Message Source	Exit Port	Destination Node
1	1	3	Node 2
1	1	2	Node 3
1	1	4	Node 4
2	1	2	Node 6
4	1	0	Node 5
5	1	2	Node 7
6	1	2	Node 9
1	2	3	Node 2
2	2	2	Node 6
1	3	4	Node 4
...	...	...	...

The code changes to accommodate message copying and distribution are shown here. This simply copies the message and sends to each output port specified in the routing file. The original message is then destroyed.

```
op_pk_nfd_get (pkptr, "source", &source);
```

```
for (i = 0;i<tg_route_rows;i++)
{
  if (tg_route[i].node == tg_node)
    if (tg_route[i].source == source)
    {
      outport = tg_route[i].outport;
      pkptr2 = op_pk_copy (pkptr);
      op_pk_send(pkptr2, outport);
    }
}
op_pk_destroy (pkptr);
```

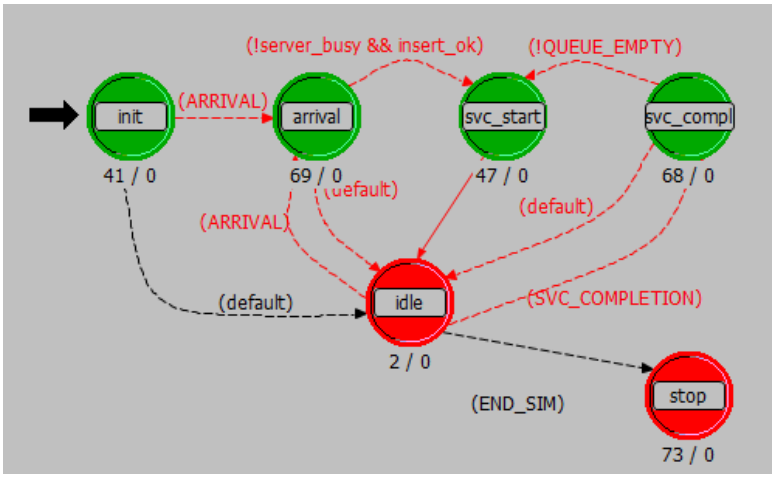


Figure 73: Routing/Forwarding Process Diagram

A.2 Message Generation Node

The message source is a typical OPNET packet generator with a modification to allow for the use of ME distributions both in the arrival process and in the message size process. A novel method to generate ME random variates is proposed Section B.6. Figures 74 and 75 show the message generation node and process models, which look the

same as the standard packet generation Modeler models. The only differences are in the code and are shown below.

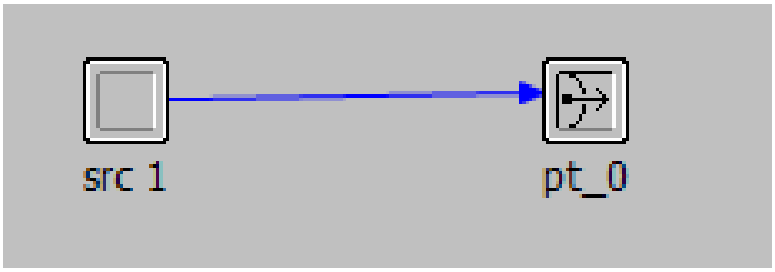


Figure 74: Message Generation Node

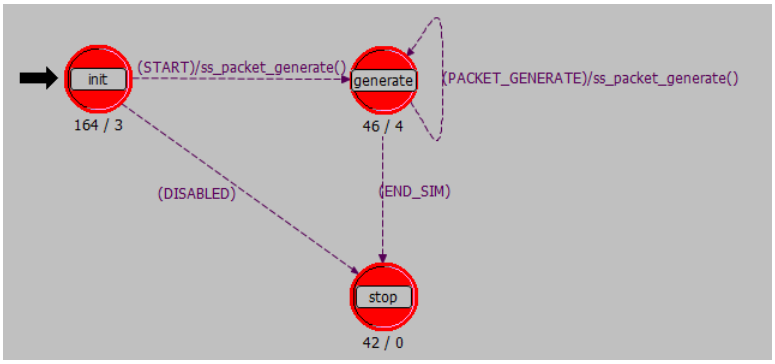


Figure 75: Message Generation Process

Configuration is via the OPNET attributes for the specific source. Figure 76 shows a typical hyper-exponential distribution being used to generate interarrival times (src 1.Arrival) with $< p, B, e >$ matrices as shown in (A.1). The B matrix is entered as a single line with columns and rows separated by spaces. The RNG code assembles the matrix from that data and the- ME_size attribute. The ME_use attribute selects exponential, hyper-exponential, erlang, or general ME distributions. This enables the RNG to select the most efficient method of generating random variates.

$$p = \begin{bmatrix} 0.788 & 0.2113 \end{bmatrix} \quad B = \begin{bmatrix} 52.57 & 0.0 \\ 0.0 & 14.01 \end{bmatrix} \quad e = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{A.1}$$

Attribute	Value
name	Source 8
src 1.Arrival B	52.57308 0 0 14.08692
src 1.Arrival M1	0.0
src 1.Arrival M2	0.0
src 1.Arrival M3	0.0
src 1.Arrival ME Size	2
src 1.Arrival ME_use	3
src 1.Arrival e	1 1
src 1.Arrival p	0.78868 0.21132
src 1.Node	8

Figure 76: Source Configuration Attributes

The code snippet modifications to allow multiple random number generators is shown here. Option 0 uses the Modeler random number generator (RNG). Option 1 uses the ME RNG. Options 2 and 3 use 4 uses Exponential, Hyper-Exponential, and Erlang RNGs. These are described in Appendix B.

```
FIN (ss_packet_generate ());

  /* Generate a packet size outcome. */
  if (tg_svc_me_use == 0)
    pktime = (double) (oms_dist_outcome (pksize_dist_ptr));
  else if (tg_svc_me_use == 1)
    pktime = svc_dist.tg_get_merv(2.0);
else if (tg_svc_me_use == 2)
pktime = svc_dist.tg_get_exp();
  else if (tg_svc_me_use == 3)
    pktime = svc_dist.tg_get_hyper2();
```

```
else if (tg_svc_me_use == 4)
  pktime = svc_dist.tg_get_erlang();
```

APPENDIX B

ME RANDOM NUMBER GENERATION

This appendix describes the Matrix Exponential (ME) Random Number Generators (RNG) that were constructed for the simulation associated with the performance analysis. Other RNGs that were constructed and used are also descibed. Finally, we describe a novel numeric linear combination approach to ME RNG.

B.1 Related Work

Brown, et al. [24] describes ME RV generation approaches that uses an accept/reject method for the tail of the ME distribution and a numerical CDF inversion via bisection for the remainder of the distributio n. This was demonstrated on several distributions and ME distributions created using moment matching of empirical samples. The moment matching techniques to create ME distributions by Van de Liefvoort in [151] are used in our work as well. The methods proposed in this work were able to generate ME RVs with one uniform random variate and on the average 19 matrix exponential calculations. It should be noted that the methods proposed in [24] had no limitations with respect to the type of ME distributions used, and this work was extended to generate RVs for correlated ME processes in 2006 by Fitzgerald, et al. [47].

For phase type distributions and certain ME distributions (those including Markovian generator matrices and *all ones* summation vectors) relatively efficient RNGs have been proposed recently that take a different approach. Horvath, Telek, and others [74] [129], have built a suite of tools known as LIBPHPRNG library that can model many ME distributions in a representative block form. These forms include the Hyper-Feedback-Erlang form and Hypo-Feedback-Erlang form. Once in this form, each block is reduced to a less complicated form using the eigenvalues of that block. If the eigenvalues are real, an Erlang block is used. If the eigenvalues are complex, a transformation is applied that enlarges the block, and eliminates the complex components. Essentially, their method constructs an efficient phase-type representation, if possible. Once in PH representation, a sequence of uniform random variates is generated, arranged in proper form, taking logarithms as necessary to create the final ME/PH random variate.

To handle negative elements in the starting vector, the authors use an accept/reject method. They build the accept probability by separating the positive and negative coefficients of the starting vector and arranging an acceptance ratio. The number of logarithms and uniform random variates needed for their methods can vary widely and are dependent on the complexity, structure, and values of the eigenvalues of the generator matrices as well as the average acceptance probabilities if negative starting vector coefficients exist. The authors were able to reduce the complexity considerably by using efficient transformations over simply *playing* the queue. These results are described in [74]. They were

able to show that these methods could be used with empirical samples and moment matching using techniques described in [148]. In addition, they extended their work to include rational arrival processes.

One related work was completed by Rideout in [130]. While it was intended to describe methods to generate random variables for distributions that are *only* known by their Laplace transform, it is interesting to note that if the tail of the distribution is sufficiently flat, different methods of CDF inversion may need to be employed to accurately capture the properties of the distribution in this region.

While all of these methods are very useful in their respective application areas, none of these methods were sufficiently general and highly efficient in their approaches to generate ME random variates for our purposes. The ME RNG in this work fulfills the need to have an efficient and general ME RNG.

Modeler includes a variety of distributions that can generate random variates for interarrival and packet size distributions. Generally while useful, they do not provide the flexibility we need in this model. Matrix Exponential models are necessary for that level of flexibility. Four random number generators were created for use with this simulation model. These are described in this section. They include Exponential, Hyper-Exponential, Erlang, Matrix-Exponential (ME). The ME RNG could create any of the other random variates, but the specific RNGs are generally more efficient. These specific RNG techniques are generally known as *playing the queue*.

An RNG object is created using a C++ include file. The structure of the object is as follows:

```
class ME_dist {
  public:
    MatrixXd B;
    MatrixXd V;
    MatrixXd p;
    MatrixXd e;
    ...
```

Where B, V, p, e are matrix objects using the Eigen library [66]. They represent the $< p, B, e >$ and V matrices in Section 3.3.

B.2 Exponential RNG

The Exponential RNG simply uses the Modeler $op\_dist\_exponential()$ function to create the Exponential RNGs.

B.3 Hyper2-Exponential RNG

The Hyper2-Exponential RNG on probability p generates an exponential random variate with mean of $1.0/B(0,0)$ and $1.0/B(1,1)$ with probability $1 - p$. The code is shown:

```
double r1, r2, p1, u, mernd;
r1 = B(0,0);
r2 = B(1,1);
p1 = p(0);
u = op_dist_uniform(1.0);
if (u < p1)
```

```
  mernd = op_dist_exponential(1.0/r1);
else
  mernd = op_dist_exponential(1.0/r2);
return(mernd);
```

B.4 Erlang RNG

The Erlang RNG generates k exponential random variates with a mean of $B(0,0)$. These are summed to create the Erlang random variate. The code is shown:

```
double sz, r, mernd;
int i;
sz = B.rows();
r = B(0,0);
mernd = 0;
for (i=0; i<sz; i++)
  mernd = mernd + op_dist_exponential(1.0/r);
return(mernd);
```

B.5 Matrix Exponential (ME) RNG

There are a couple of methods of generating ME RNGs. The *playing the queue* techniques can be extended to include hyper-erlang variations. They still cannot generate many ME variations.

Other methods include numerical CDF Inversion Methods. This work uses a variation of that method here. Coupled with a table lookup and interpolation, this creates an efficient and accurate RNG. In addition, this work pioneered another variation of this work that contains the maximum number in the lookup table between 0 and 1 considerably simplifying the lookup process. This is shown in Section B.6.

The *ME Lookup* method involves a couple of steps. The table is created using a numerical CDF inversion while sampling from 0.0 to 1.0. Random Variates are created by generating uniform random variates between 0.0 and 1.0. These are then used in the table lookup. The code used for numerical CDF inversion is shown:

```
double ME_dist::tg_rand_me(double u)
{
  int sz, j, stat, itr;
  double a, b, x, r, dtem1, dtem2;
  sz = B.rows();
  MatrixXd Mtem1(1,1), Mtem2(sz,sz), Mtem3(sz,sz);
  b = rvmax;  // max value of random variable
  // initialize
  a = 0;
  itr = 0;
  stat = 0;
  j = 0;

  // start loop
  while (stat == 0)
  {
    j = j + 1;
    itr = itr + 1;
    x = (a+b)/2;
    Mtem2 = -B*x;
    Mtem3 = Mtem2.exp();
    Mtem1 = p*Mtem3*e;
    dtem1 = (double) Mtem1(0,0);
    dtem2 = 1.0 - dtem1;

    if (dtem2 <= u)
      a = x;
    else
      b = x;

    if ((b-a) < err)
```

```
        stat = 1;
    }
    return(x);
}
```

B.6 ME RNG using a Numeric Linear Combination Approach

The goal of this ME RNG is to create a simple, accurate, and efficient ME random number generator (RNG) that could be used for a wide variety of ME distributions including those with complex and negative components in the constituent matrices. Scalability is important because of the state space growth associated with certain ME queueing operations. It also became clear that *closed* form solutions were not feasible and numerical solutions may present the best options to achieve these goals.

A challenge of random number generation is capturing the properties of the tail of the distribution correctly. Several works [38, 48, 50] have documented the existence of self-similar and heavy tailed distributions in internet traffic, specifically file sizes in web traffic. A review of ME Distributions is in Section 2.1. An advantage of ME distributions is that heavy tailed and self-similar distributions can be modeled as is demonstrated by Lipsky in [91]. The methods described this current work can be used to generate random numbers for these distributions (within the capability of the simulation model). A feature of this RNG is that the entire tail of the distribution need not be modeled.

In the work by Felgueiras et al. [45], the relationship between distribution functions and density functions that are closed under minimization is exploited to create non-convex mixtures of the base density function. Depending on the density function chosen, the corresponding mixture function may be composed of a weighted sum of the base density function and square of the base density function. In this case, the mixture distribution is re-written and the base density function is replaced with a variable. The new equation is solved in closed form using the quadratic equation. If the base density function can be inverted, random variates can readily be generated. An example using the exponential distribution was shown.

Using a similar approach and noting $e^{-ax} = (e^{-x})^a$, the ME CDF for e^{-x} is solved numerically which allows a uniform random variate to be scaled for the particular ME distribution prior to an exponential inversion. To accomplish this, the ME distribution is transformed to diagonal form, which enables the distribution to be written as a *linear combination* of exponentials and the equation is solved numerically. In this work, we relax the requirement that the weights are positive. Since they still sum to one, the density is non-negative, and that the density integrates to one; we are only working with valid distributions. The advantage of this approach is that the random variates are generated from a single uniform random variate, using a single logarithm calculation, and using a single interpolation. This makes the RNG efficient and accurate. The disadvantage of this approach is the requirement to store numerical information about the scaling function.

We will demonstrate the flexibility, efficiency, and accuracy of this approach.

B.6.1 Development of the ME RNG

B.6.1.1 ME Random Variate Generation

To create random variates for a given ME distribution, $\langle \mathbf{p}, \mathbf{B}, \mathbf{e} \rangle$, we start with the $N \times N$ size $\mathbf{B}$ matrix. If the matrix $\mathbf{B}$ is not defective, matrix decomposition can be used to diagonalize $\mathbf{B}$ (defective $\mathbf{B}$ matrices are discussed later in this section). The CDF can be rewritten as shown in (B.2) where $\mathbf{B}^*$ is a diagonal matrix with the eigenvalues $(\lambda_1, \lambda_2, ...)$ on the diagonal. λ_i and the eigenvector matrix $\mathbf{X}$ can be complex and/or negative, which is not a concern if (B.1) is true ensuring that $f(x)$ is a valid density function.

$$f(x) \geq 0 \quad \forall x > 0 \quad \text{and} \quad \int_0^\infty f(x)dx = 1 \tag{B.1}$$

$$F(x) = 1 - \mathbf{p} exp(-\mathbf{B}x)\,\mathbf{e}' = 1 - \mathbf{p}\mathbf{X} exp(-\mathbf{B}^* x)\,\mathbf{X}^{-1}\mathbf{e}' \tag{B.2}$$

Re-arranging into a new representation of the original ME distribution, the new $\mathbf{p}^*$ and $\mathbf{e}^*$ are shown in (B.3) which leaves (B.4).

$$\mathbf{p}^* = \mathbf{p}\mathbf{X}, \quad \mathbf{e}^{*\prime} = \mathbf{X}^{-1}\mathbf{e}' \tag{B.3}$$

$$F(x) = 1 - \mathbf{p}^* exp(-\mathbf{B}^* x)\,\mathbf{e}^{*\prime} \tag{B.4}$$

It is clear that the CDF is a linear combination of exponentials as shown in (B.5) where α_i is $p_i^* \times e_i^*$. The exponentials λ_i is element (i,i) of $\mathbf{B}^*$ where $\mathbf{B}^*$ is of size NxN. α and λ can be complex valued, in which case a complex conjugated peer component exist which cancels the purely complex values.

$$F(x) = 1 - (\alpha_1 e^{-\lambda_1 x} + \alpha_2 e^{-\lambda_2 x} + ... + \alpha_N e^{-\lambda_N x}) \tag{B.5}$$

There are specific ME formulations that may have defective $\mathbf{B}$ matrices, like the following 4×4 Erlang ME distribution (B.6). It is clear that $\mathbf{B}$ is defective and would yield an eigenvector matrix that was conditioned poorly. Noting the PDF of this distribution in (B.7), we can see that we have repeated eigenvalues.

$$B = \lambda \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.6}$$

$$f\left(x; k, \lambda\right) = \frac{\lambda}{(k-1)!}(\lambda x)^{k-1} e^{-\lambda x} \quad x > 0 \tag{B.7}$$

In order to keep this RNG as general as possible and not make exceptions for a particular ME formulation, a matrix perturbation method is used to condition the matrix as necessary. Generally, if the $\mathbf{B}$ fails a standard eigenvalue condition test, a small negative random perturbation (order of magnitude is 10^{-9}) is added to the lower left corner of the $\mathbf{B}$ matrix. If the the matrix is still defective, the perturbation is increased. We were able to

capture the properties of the Erlang distribution closely using this conditioning technique. For a more in depth review of eigenvalue conditioning, the reader is referred to Wilkinson in [156, p. 90].

The classic method of generating random variates of a distribution $f(x)$ is to generate a uniform random variate and set it equal to the CDF, $F(x) = U$ The new random variate is obtained by solving for x, $x = F^{-1}(U)$. Applying this to (B.5) we now have (B.8):

$$U = 1 - (\alpha_1 e^{-\lambda_1 x} + \alpha_2 e^{-\lambda_2 x} + ... + \alpha_N e^{-\lambda_N x}) \tag{B.8}$$

Since $1 - U$ is also a uniform distribution on $U(0, 1)$, $1 - U$ can be replaced with U. The exponentials can now be rewritten as $(e^{-x})^{\lambda_i}$.

$$U = \alpha_1 (e^{-x})^{\lambda_1} + \alpha_2 (e^{-x})^{\lambda_2} + ... + \alpha_N (e^{-x})^{\lambda_N}) \tag{B.9}$$

In a manner similar to [45], we substitute e^{-x} with z leading to (B.10).

$$U = (\alpha_1 z^{\lambda_1} + \alpha_2 z^{\lambda_2} + ... + \alpha_N z^{\lambda_N}) \tag{B.10}$$

Next, $g(z)$ is defined as shown in (B.11), which leads to (B.12). In this formulation, α and λ can be negative and/or complex. Since complex components are in conjugate pairs, the purely complex value is canceled.

$$g(z) = (\alpha_1 z^{\lambda_1} + \alpha_2 z^{\lambda_2} + ... + \alpha_N z^{\lambda_N}) = \sum_{i=1}^{N} \alpha_i z^{\lambda_i} \tag{B.11}$$

$$U = g(z), \qquad z = g^{-1}(U) \tag{B.12}$$

Since $e^{-x} = z$, x can be solved with a simple logarithm (B.13).

$$x = -log(z) \tag{B.13}$$

Since $g^{-1}(U)$ is difficult to solve in closed form, we generate a $g^{-1}(u)$ numerically as u goes from 0 to 1 and perform an linear interpolation to arrive at z. Since $g^{-1}(u)$ is generated prior to random number generation, each random number requires a single uniform random variate, interpolation, and logarithmic calculation.

Theorem 1. *Equation* (B.11) *has a single unique solution in the range of [0,1].*

Proof. Since $g(z)$ in (B.11) is $1 - F(x)$, $g(z)$ is by definition monotonically decreasing. At $z = 0$, it is clear that $g(z) = 0$. At $z = 1$, $g(z) = 1$. We can see this by letting x go to zero in (B.5). $F(x)$ also must go to zero therefore $\alpha_1 + \alpha_2 + ... + \alpha_N$ must be one and therefore $g(1) = 1$. Finally, since the above are true, there can only be one possible solution to (B.11) in the range [0,1]. □

B.6.1.2 ME RNG Algorithms

Algorithm 13 shows the ME RNG setup procedure. The initial setup consists of eigenvalue decomposition, similarity transformation, and recombination into a linear combination of exponentials. After the linear combination of exponentials is created, $g^{-1}(u)$ is calculated numerically using a standard bisection algorithm where M is the number of points representing $g^{-1}(u)$.

Algorithm 13 ME RNG Setup

$\lambda_i \leftarrow$ Eigenvalues of B
$X \leftarrow$ Eigenvectors of B
$p^* \leftarrow pX$
$e^{*\prime} \leftarrow X^{-1}e'$
$\alpha_i \leftarrow p_i^* \times e^{*\prime}_i$
$g(x) \leftarrow \alpha_1 x^{\lambda_1} + ... + \alpha_N x^{\lambda_N}$
for all i such that $0 \leq i \leq M$ **do**
  $u_i \leftarrow \frac{i}{M}$
  $z_i \leftarrow g^{-1}(u_i)$ using bisection
end for

Algorithm 14 shows the ME random variate generation procedure. To generate an ME random variate, the following steps are followed. First, a uniform random variate is generated between $[0, 1]$. Next, $z = g^{-1}(U)$ is calculated using simple interpolation. The ME random variate is calculated by taking the logarithm of z and multiplying by -1.

Algorithm 14 ME RV Generation

$U \leftarrow$Generate Uniform Random Variate
$z \leftarrow g^{-1}(U)$ using linear interpolation
ME RV $\leftarrow -log(z)$

These algorithms have been implemented in C++ and Octave [115]. In C++, the Eigen Library [66] was used for Matrix Exponential Computations and Eigenvalue decomposition. The C++ implementation has been imported into Opnet [118] using C++ include files. We have used the ME RVs as both arrival and service distributions during simulation of a $Gi/G/1$ queue which is a part of ongoing research.

B.6.1.3 Complexity

The process of generating an ME RV involves a single ME RNG setup and multiple ME RV Generations. The setup costs (in time) vary depending on the distribution and are generally trivial compared to the ME RV generation for large runs of random variants. Using an Intel Core I-7 with 2.4 GHz processors, the setup costs range from approximately 250ms for a typical distribution to several seconds for a heavy tail distribution. Is should be noted that this is a one time cost per distribution.

Since ME random variate requires a single uniform random variate, interpolation, and logarithmic calculation, the process to generate ME random variates is fast. Table 26 shows the number of ME RVs per second versus the number of exponential RVs per second that were generated using the C++ TR1 Random Library.

Table 26: ME RNG Performance

	RVs/sec
Exponential	4.1×10^6
ME	3.7×10^6

B.6.1.4 Numerical Considerations

Eigenvalue decomposition and subsequent matrix inversion can lead to numerical problems and we refer to Section 2.2 for a more complete discussion of the Eigenvalue decomposition approach used in this work.

The selection of M affects the precision of the numerical calculation of $z = g^{-}1(u)$. ME PDFs have two regions of concern. First the main body of the PDF can generally be captured using a relatively small number of data points. $M = 1000$ was used in this work. The tail of the distribution is reliant on the region where u is near zero. Depending on the distribution, $g^{-1}(u)$ can be very small near this region. In order to calculate the RVs on the tail of the distribution, $g^{-1}(u)$ needs to be estimated accurately in this region. This is primarily a concern with heavy tailed distributions. Two measures were used to mitigate these concerns.

First, the stopping error in the bisection algorithm, was set to 1×10^{-6}. But, if $z < 10 \times error$ the error was reduced by a factor of 10 during the bisection. This is continued as necessary till the error is near the overall smallest possible value of the system being used (1×10^{-200} was used in this work). Second, if $z_i/z_{i-1} > 10$, M is increased by a factor of 100 for that region. This enabled a better linear approximation of that region.

This approach to generate ME random variates provides efficient and accurate RVs for a wide variety of probability distributions.

B.6.2 Examples

We tested the ME RNG with a wide variety of distributions including exponential and hyper-exponential. For brevity, we chose to show the more interesting examples that demonstrate the efficiency, flexibility, and accuracy of the ME RNG. The examples include Erlang, complex sinusoidal, moment matched and heavy tail distributions.

B.6.2.1 Erlang Case

The first example is the Erlang distribution. Equation (B.14) shows the $\langle \mathbf{p}, \mathbf{B}, \mathbf{e} \rangle$ representation for a $k = 6$ size Erlang distribution. Figure 77 shows the $g^{-1}(u)$ function for the Erlang-6 distribution. Table 27 shows the $g(z)$ parameters. Figure 78 shows the PDF and histogram of the generated RVs. Table 28 shows the calculated moments versus the moments generated from the RNG with 1×10^7 RVs generated. A random perturbation on the order of -1×10^{-4} was added to $B(6,1)$ location for this case (see discussion of Erlang distributions in Section B.6.1.1).

$$p = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad e = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{B.14}$$

Table 27: Erlang $g(z)$ Parameters

N	α_i	λ_i
1	457.63 + 0.0i	0.7846 + 0.0i
2	122.95 - 374.18i	0.8923 + 0.1866i
3	122.95 + 374.18i	0.8923 - 0.1866i
4	-203.56 - 246.41i	1.1077 + 0.1866i
5	-203.56 + 246.41i	1.1077 - 0.1866i
6	-295.40 + 0.0i	1.2154 - 0.0i

Table 28: Erlang Distribution Performance

	Calculated	RV Generated
1st Moment	6.00	6.00
2nd Moment	42.00	42.01
3rd Moment	336.00	336.18
C^2	0.17	0.17

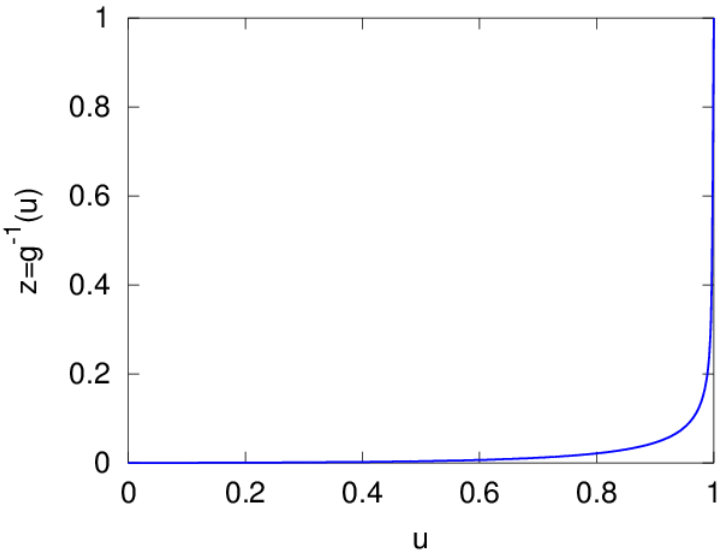


Figure 77: Erlang $g^{-1}(u)$ Function

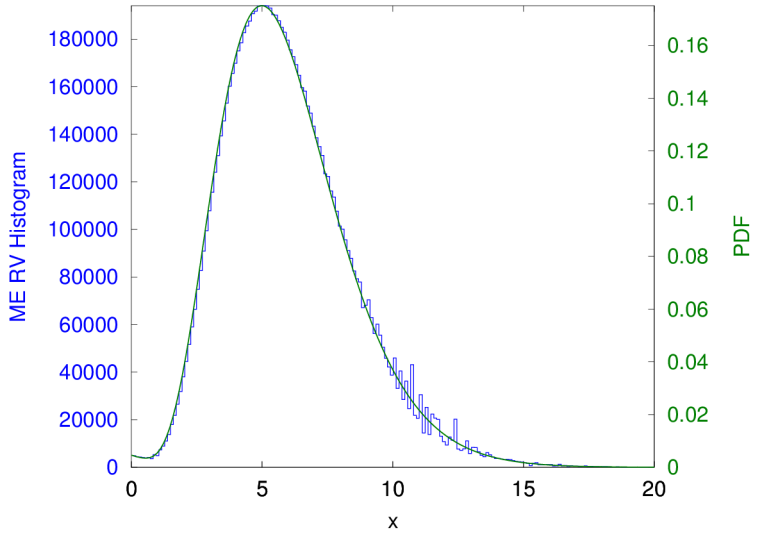


Figure 78: Erlang ME RV Histogram and PDF

B.6.2.2 Sinusoidal Case

The second example is a distribution with a strong sinusoidal component in the PDF. Equation (B.15) shows the $\langle \mathbf{p}, \mathbf{B}, \mathbf{e} \rangle$ representation for this distribution. Figure 79 shows the $g^{-1}(u)$ function for the sinusoidal distribution. 29 shows the $g(z)$ parameters. Figure 80 shows the PDF and histogram of the generated RVs. Table 30 shows the calculated moments versus the moments generated from the RNG with 1×10^7 RVs generated.

$$p = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.2222 & -1.2222 & 0 \\ 0 & 1.2222 & -1.2222 \\ 0 & 9.7778 & 1.2222 \end{bmatrix} \quad e = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{B.15}$$

Table 29: Sinusoidal $g(z)$ Parameters

N	α_i	λ_i
1	1.125000 + 0.000000i	1.2222 + 0.0000i
2	-0.062500 - 0.176777i	1.2222 + 3.4570i
3	-0.062500 + 0.176777i	1.2222 - 3.4570i

Table 30: Sinusoidal PDF Performance

	Calculated	RV Generated
1st Moment	1.00	1.00
2nd Moment	1.55	1.55
3rd Moment	3.69	3.69
C^2	0.554	0.554

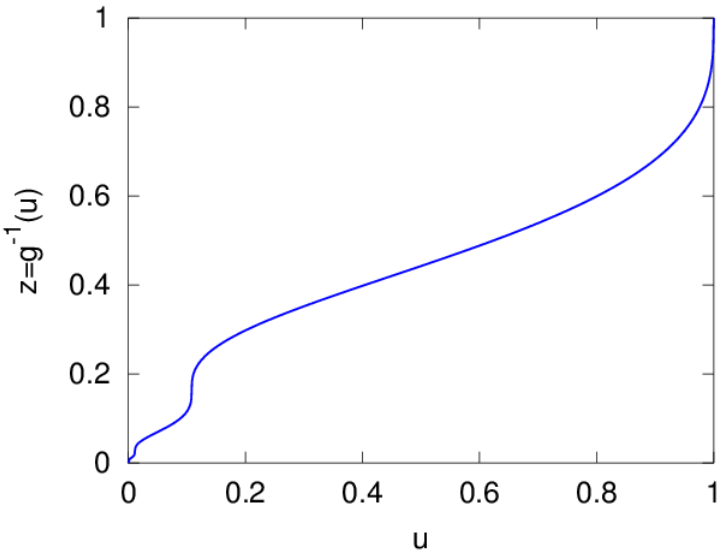


Figure 79: Sinusoidal $g^{-1}(u)$ Function

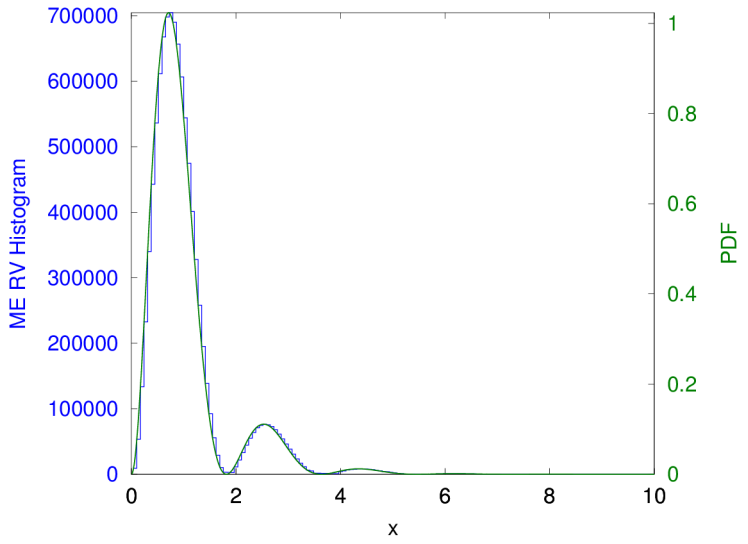


Figure 80: Sinusoidal ME RV Histogram and PDF

B.6.2.3 Empirical Sample using a Moment Match

This example is a distribution constructed with a moment matching technique as noted in [151]. Equation (B.16) shows the $\langle \mathbf{p}, \mathbf{B}, \mathbf{e} \rangle$ representation for this distribution. Figure 81 shows the $g^{-1}(u)$ function for the moment matched distribution. 31 shows the $g(z)$ parameters.Figure 82 shows the PDF and histogram of the generated RVs. Table 32 shows the calculated moments versus the moments generated from the RNG with 1×10^7 RVs generated.

$$p = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 2.0 & -1.0 \\ -1.0 & 1.0 \end{bmatrix} \quad e = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{B.16}$$

Table 31: Moment Matched $g(z)$ Parameters

N	α_i	λ_i
1	0.2764	0.3820
2	0.72361	2.6180

Table 32: Moment Generated PDF Performance

	Design	Calculated	RV Generated
1st Moment	1	1.0	1.00
2nd Moment	4	4.0	3.99
3rd Moment	30	30.0	29.94
C^2		3.00	3.00

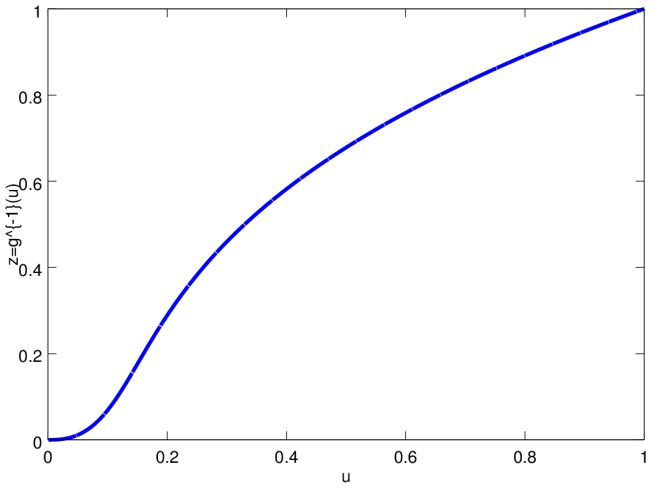


Figure 81: Moment Matched $g^{-1}(u)$ Function

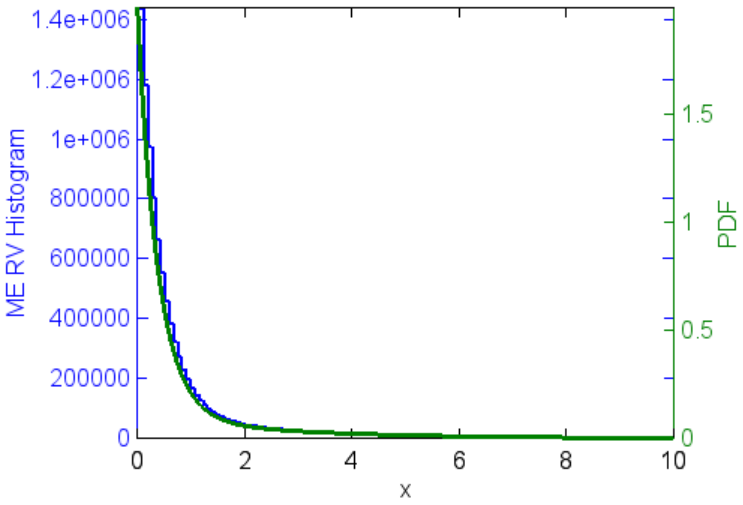


Figure 82: Moment Matched ME RV Histogram and PDF

B.6.2.4 Truncated Power Tail Case

The final example is a distribution with a heavy tail component in the PDF. This example of a Truncated Power Tail (TPT) distribution can be found in [91] by Lipsky. Equation (B.17) shows the $\langle \mathbf{p}, \mathbf{B}, \mathbf{e} \rangle$ representation for this distribution. 33 shows the $g(z)$ parameters. Figure 83 shows the $g^{-1}(u)$ function for the sinusoidal distribution. Figure 84 shows the PDF and histogram of the generated RVs. Table 34 shows the calculated moments versus the moments generated from the RNG with 1×10^7 RVs generated.

$$p = \begin{bmatrix} 0.70 & 0.21 & 0.063 & 0.019 & 0.0057 & 0.0017 & 0.0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 & -1.0 \\ 0 & 0.423 & 0 & 0 & 0 & 0 & -0.423 \\ 0 & 0 & 0.179 & 0 & 0 & 0 & -0.179 \\ 0 & 0 & 0 & 0.0758 & 0 & 0 & -0.0758 \\ 0 & 0 & 0 & 0 & 0.0321 & 0 & -0.0321 \\ 0 & 0 & 0 & 0 & 0 & 0.0136 & -0.0136 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.010 \end{bmatrix} \quad e = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{B.17}$$

Table 33: TPT $g(z)$ Parameters

N	α_i	λ_i
1	-0.007076	1.0000
2	-0.005086	0.4232
3	-0.003729	0.1791
4	-0.002875	0.07578
5	-0.002571	0.03207
6	-0.004768	0.01357
7	1.02611	0.01000

The TPT distribution (and any heavy tail distribution) can cause numerical issues due to the extremely small probability of large mass items occurring. We see this in Fig. 84 with the RVs that are generated at $x = 700$. Noting that $e^{-700} \approx 1 \times 10^{-304}$ is the limit

Table 34: TPT PDF Performance

	Calculated	RV Generated
1st Moment	102.10	101.96
2nd Moment	20464	20281
3rd Moment	6.1447e6	5.9406e6
C^2	.963	.951

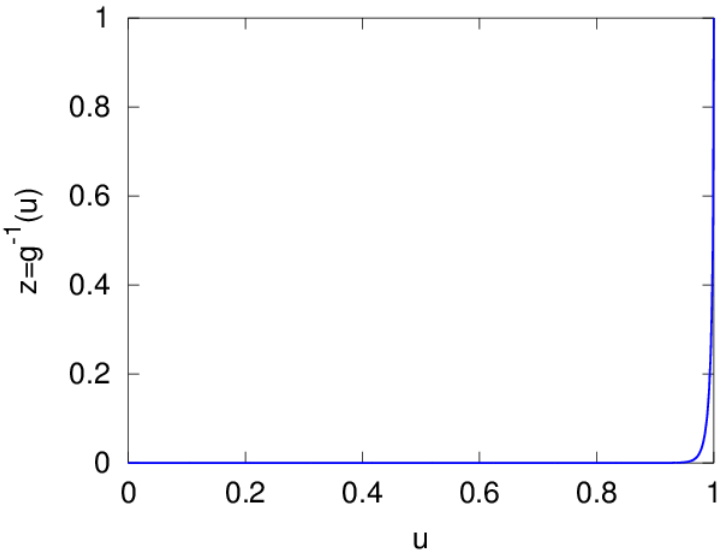


Figure 83: TPT $g^{-1}(u)$ Function

of precision on most machines. Even with this anomaly, the RVs generated were still high quality based on the moments and PDF. It should be noted that heavy tail and rare event simulation required careful analysis of the distributions to ensure that numerical errors do not become significant in RV generation.

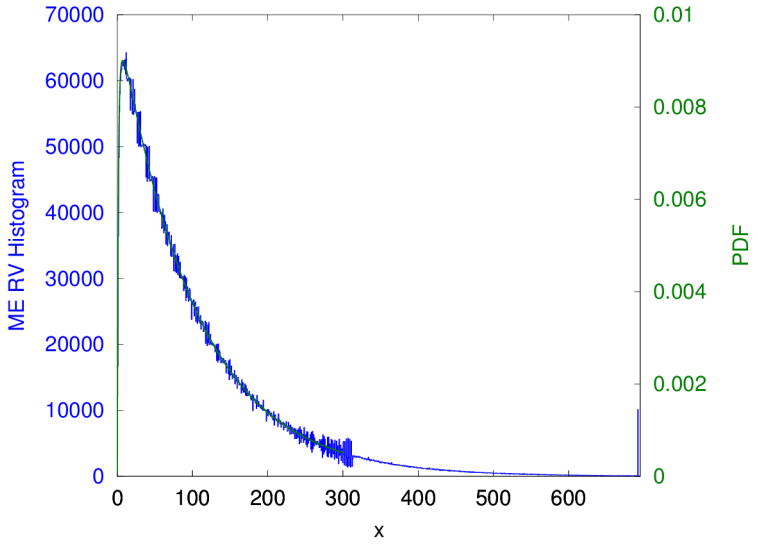


Figure 84: TPT ME RV Histogram and PDF

REFERENCE LIST

[1] Agarwal, P. K., Efrat, A., Ganjugunte, S. K., Hay, D., Sankararaman, S., and Zussman, G. Network vulnerability to single, multiple, and probabilistic physical attacks. In *MILITARY COMMUNICATIONS CONFERENCE, 2010-MILCOM 2010* (2010), IEEE, pp. 1824–1829.

[2] Agarwal, P. K., Efrat, A., Ganjugunte, S. K., Hay, D., Sankararaman, S., and Zussman, G. The resilience of WDM networks to probabilistic geographical failures. *IEEE/ACM Transactions on Networking (TON) 21*, 5 (2013), 1525–1538.

[3] Ahn, Y.-Y., Han, S., Kwak, H., Moon, S., and Jeong, H. Analysis of topological characteristics of huge online social networking services. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 835–844.

[4] Airlines for America (A4A). Per-minute cost of delays to U.S. airlines. http://airlines.org/data/per-minute-cost-of-delays-to-u-s-airlines/, 2014. [Online; accessed 6-April-2016].

[5] Akaike, H. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on 19*, 6 (1974), 716–723.

[6] Albin, S. L. Approximating a point process by a renewal process, II: Superposition arrival processes to queues. *Operations Research 32*, 5 (1984), 1133–1162.

[7] Alenazi, M. J., Çetinkaya, E. K., and Sterbenz, J. P. Cost-efficient algebraic connectivity optimisation of backbone networks. *Optical Switching and Networking 14* (2014), 107–116.

[8] Alenazi, M. J., Cetinkaya, E. K., and Sterbenz, J. P. Cost-Efficient network improvement to achieve maximum path diversity. In *Reliable Networks Design and Modeling (RNDM), 2014 6th International Workshop on* (2014), IEEE, pp. 202–208.

[9] Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., and Xu, M. Web services agreement specification (WS-Agreement). In *Open Grid Forum* (2007), vol. 128.

[10] Araghi, M., et al. A new renewal approximation for certain autocorrelated processes. *Operations Research Letters 36*, 1 (2008), 133–139.

[11] Badidi, E. A framework for brokered service level agreements in SOA environments. In *Next Generation Web Services Practices (NWeSP), 2011 7th International Conference on* (2011), IEEE, pp. 37–42.

[12] Badidi, E., Esmahi, L., and Serhani, M. A. A queuing model for service selection of multi-classes QoS-aware web services. In *Web Services, 2005. ECOWS 2005. Third IEEE European Conference on* (2005), IEEE.

[13] Ball, M. O., Colbourn, C. J., and Provan, J. S. Network reliability. *Handbooks in operations research and management science 7* (1995), 673–762.

[14] Banerjee, S., Shirazipourazad, S., and Sen, A. Design and analysis of networks with large components in presence of region-based faults. In *Communications (ICC), 2011 IEEE International Conference on* (2011), IEEE, pp. 1–6.

[15] Bardhan, S., and Milojicic, D. A mechanism to measure quality-of-service in a federated cloud environment. In *Proceedings of the 2012 workshop on Cloud services, federation, and the 8th open cirrus summit* (2012), ACM, pp. 19–24.

[16] Basu, A., and Riecke, J. Stability issues in OSPF routing. *ACM SIGCOMM Computer Communication Review 31*, 4 (2001), 225–236.

[17] Bellavista, P., Corradi, A., and Reale, A. Quality of service in wide scale publish subscribe systems. *Communications Surveys Tutorials, IEEE 16*, 3 (Third 2014), 1591–1616.

[18] Berbner, R., Spahn, M., Repp, N., Heckmann, O., and Steinmetz, R. Heuristics for QoS-aware web service composition. In *Web Services, 2006. ICWS'06. International Conference on* (2006), IEEE, pp. 72–82.

[19] Berkelaar, M., Eikland, K., and Notebaert, P. lp_solve 5.5. 0.10. http://lpsolve. sourceforge. net/5.5, 2007. [Online; accessed 6-April-2016].

[20] Bianco, P., Lewis, G. A., and Merson, P. Service level agreements in service-oriented architecture environments. Tech. rep., Defense Technical Information Center (DTIC), September 2008. [Online: http://www.dtic.mil/get-tr-doc/pdf?AD=ADA528751].

[21] Bigdeli, A., Tizghadam, A., and Leon-Garcia, A. Comparison of network criticality, algebraic connectivity, and other graph metrics. In *Proceedings of the 1st Annual Workshop on Simplifying Complex Network for Practitioners* (2009), ACM, p. 4.

[22] Bird, B. Fire at Aurora FAA facility exposed 'vulnerabilities,' says report. *Chicago Tribune* (2015). [Online; accessed 6-April-2016].

[23] Bishop, C. M., et al. *Pattern recognition and machine learning*, vol. 1. springer New York, 2006.

[24] Brown, E., Place, J., and van de Liefvoort, A. Generating matrix exponential random variates. *Simulation 70*, 4 (1998), 224–230.

[25] Burnham, K. P., and Anderson, D. R. Multimodel inference understanding AIC and BIC in model selection. *Sociological methods & research 33*, 2 (2004), 261–304.

[26] Canfora, G., Di Penta, M., Esposito, R., and Villani, M. L. An approach for QoS-aware service composition based on genetic algorithms. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation* (2005), ACM, pp. 1069–1075.

[27] Cascella, R. G., Blasi, L., Jegou, Y., Coppola, M., and Morin, C. *The Future Internet: Future Internet Assembly 2013: Validated Results and New Horizons*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, ch. Contrail: Distributed Application Deployment under SLA in Federated Heterogeneous Clouds, pp. 91–103.

[28] Cetinkaya, E., Alenazi, M., Cheng, Y., Peck, A., and Sterbenz, J. On the fitness of geographic graph generators for modelling physical level topologies. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2013 5th International Congress on* (Sept 2013), pp. 38–45.

[29] Cetinkaya, E. K., Alenazi, M. J., Cheng, Y., Peck, A. M., and Sterbenz, J. P. A comparative analysis of geometric graph models for modelling backbone networks.

Optical Switching and Networking 14, Part 2 (2014), 95 – 106. Special Issue on (RNDM) 2013.

[30] Cetinkaya, E. K., Broyles, D., Dandekar, A., Srinivasan, S., and Sterbenz, J. P. A comprehensive framework to simulate network attacks and challenges. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on* (2010), IEEE, pp. 538–544.

[31] Cheng, Y., Gardner, M. T., Li, J., May, R., Medhi, D., and Sterbenz, J. Optimised heuristics for a geodiverse routing protocol. In *Design of Reliable Communication Networks (DRCN'2014), 2014 10th International Conference on the* (April 2014), IEEE, pp. 1–9.

[32] Cheng, Y., Gardner, M. T., Li, J., May, R., Medhi, D., and Sterbenz, J. P. Analysing GeoPath diversity and improving routing performance in optical networks. *Computer Networks 82* (2015), 50–67.

[33] Cholda, P., Tapolcai, J., Cinkler, T., Wajda, K., and Jajszczyk, A. Quality of resilience as a network reliability characterization tool. *Network, IEEE 23*, 2 (2009), 11–19.

[34] Cicic, T. On basic properties of fault-tolerant multi-topology routing. *Computer Networks 52*, 18 (2008), 3325 – 3341.

[35] Cicic, T., Hansen, A., Kvalbein, A., Hartmann, M., Martin, R., Menth, M., Gjessing, S., and Lysne, O. Relaxed multiple routing configurations: IP fast reroute for single and correlated failures. *IEEE Transactions on Network and Service Management 6*, 1 (2009), 1–14.

[36] Cisco. Cisco web site, OSPF shortest path first throttling. http://www.cisco.com/c/en/us/td/docs/ios/12_2s/feature/guide/fs_spftrl.html, 2012. [Online; accessed 30-October-2013].

[37] Colbourn, C. J. *Telecommunications Network Planning*. Springer US, Boston, MA, 1999, ch. Reliability Issues In Telecommunications Network Planning, pp. 135–146.

[38] Crovella, M. E., and Bestavros, A. Self-similarity in World Wide Web traffic: Evidence and possible causes. *Networking, IEEE/ACM Transactions on 5*, 6 (1997), 835–846.

[39] Department of Transportation Office of Inspector General. FAA's progress and challenges in meeting FTI transition goals Federal Aviation Administration report number AV-2008-089, 2008. [Online; accessed 6-April-2016].

[40] Diaz, O., Xu, F., Min-Allah, N., Khodeir, M., Peng, M., Khan, S., and Ghani, N. Network survivability for multiple probabilistic failures. *IEEE Communications Letters 16*, 8 (2012), 1320–1323.

[41] Dotson, W., and Gobien, J. A new analysis technique for probabilistic graphs. *Circuits and Systems, IEEE Transactions on 26*, 10 (1979), 855–865.

[42] Esmaeili, M., Peng, M., Khan, S., Finochietto, J., Jin, Y., and Ghani, N. Multi-domain DWDM network provisioning for correlated failures. In *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2011 and the National Fiber Optic Engineers Conference* (2011), IEEE, pp. 1–3.

[43] Fay, D., Haddadi, H., Thomason, A., Moore, A. W., Mortier, R., Jamakovic, A., Uhlig, S., and Rio, M. Weighted spectral distribution for internet topology analysis: theory and applications. *Networking, IEEE/ACM Transactions on 18*, 1 (2010), 164–176.

[44] Federal Aviation Administration. National Airspace System (NAS) Overview. https://www.faa.gov/air_traffic/technology/cinp/fti2/documents/media/nas_overview.pdf, 2015. [Online; accessed 6-April-2016].

[45] Felgueiras, M., Martins, J., and Santos, R. Pseudo-convex mixtures. In *AIP Conference Proceedings* (2012), vol. 1479, p. 1125.

[46] Fendick, K. W., Saksena, V. R., and Whitt, W. Dependence in packet queues. *Communications, IEEE Transactions on 37*, 11 (1989), 1173–1183.

[47] Fitzgerald, S., Place, J., and van de Liefvoort, A. Generating correlated matrix exponential random variables. *Advances in Engineering Software 37*, 2 (2006), 75–84.

[48] Fraleigh, C., Moon, S., Lyles, B., Cotton, C., Khan, M., Moll, D., Rockell, R., Seely, T., and Diot, S. Packet-level traffic measurements from the Sprint IP backbone. *Network, IEEE 17*, 6 (2003), 6–16.

[49] Francois, P., Filsfils, C., Evans, J., and Bonaventure, O. Achieving sub-second IGP convergence in large IP networks. *ACM SIGCOMM Computer Communication Review 35*, 3 (2005), 35–44.

[50] Frost, V. S., and Melamed, B. Traffic modeling for telecommunications networks. *Communications Magazine, IEEE 32*, 3 (1994), 70–81.

[51] Fumagalli, A., and Valcarenghi, L. IP restoration vs. WDM protection: is there an optimal choice? *Network, IEEE 14*, 6 (2000), 34–41.

[52] Garbin, D., and Shortle, J. Measuring resilience in network-based infrastructures. In *CIPP, Working Paper 12-06* (Arlington, VA, 2006), George Mason University.

[53] Gardner, M. T., and Beard, C. Evaluating geographic vulnerabilities in networks. In *Communications Quality and Reliability (CQR), 2011 IEEE International Workshop Technical Committee on* (2011), IEEE, pp. 1–6.

[54] Gardner, M. T., and Beard, C. Using QNA to evaluate parameter tuning in mission critical SOA networks. In *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2012 IEEE 20th International Symposium on* (2012), IEEE, pp. 513–515.

[55] Gardner, M. T., Beard, C., and Medhi, D. Avoiding high impacts of geospatial events in mission critical and emergency networks using linear and swarm optimization. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2012 IEEE International Multi-Disciplinary Conference on* (2012), IEEE, pp. 264–271.

[56] Gardner, M. T., Beard, C., and Medhi, D. Using network measure to reduce state space enumeration in resilient networks. In *Design of Reliable Communication Networks (DRCN), 2013 9th International Conference on the* (2013), IEEE, pp. 250–257.

[57] Gardner, M. T., Beard, C., and van de Liefvoort, A. Efficient matrix-exponential random variate generation using a numeric linear combination approach. In *Proceedings of the Symposium on Theory of Modeling & Simulation-DEVS Integrative* (2014), Society for Computer Simulation International, p. 19.

[58] Gardner, M. T., Beard, C., and Van de Liefvoort, A. Mission critical publish-subscribe performance modeling using linear algebraic and classical methods. In

Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium (2015), Society for Computer Simulation International, pp. 269–277.

[59] Gardner, M. T., Cheng, Y., May, R., Beard, C., Sterbenz, J., and Medhi, D. Creating network resilience against disasters using service level agreements. In *2016 12th International Conference on the Design of Reliable Communication Networks (DRCN'2016)* (Paris, France, March 2016), IEEE.

[60] Gardner, M. T., May, R., Beard, C., and Medhi, D. Using multi-topology routing to improve routing during geographically correlated failures. In *2014 10th International Conference on the Design of Reliable Communication Networks (DRCN'2014)* (Ghent, Belgium, April 2014), IEEE.

[61] Gardner, M. T., May, R., Beard, C., and Medhi, D. Finding geographic vulnerabilities in multilayer networks using reduced network state enumeration. In *Design of Reliable Communication Networks (DRCN), 2015 11th International Conference on the* (2015), IEEE, pp. 49–56.

[62] Gardner, M. T., May, R., Beard, C., and Medhi, D. A geographic multi-topology routing approach and its benefits during large-scale geographically correlated failures. *Computer Networks 82* (2015), 34 – 49. Robust and Fault-Tolerant Communication Networks.

[63] Gardner, M. T., May, R., Beard, C., and Medhi, D. Determing geographic vulnerabilities using a novel impact based resilience metric [conditionally accepted]. *Journal of Network and Systems Management* (2016).

[64] Gerstel, O., and Sasaki, G. H. Quality of protection (QoP): a quantitative unifying paradigm to protection service grades. In *OptiComm 2001: Optical Networking and Communications Conference* (2001), International Society for Optics and Photonics, pp. 12–23.

[65] Gomes, T., Craveirinha, J., and Martins, L. An efficient algorithm for sequential generation of failure states in a network with multi-mode components. *Reliability Engineering & System Safety 77*, 2 (2002), 111–119.

[66] Guennebaud, G., Jacob, B., et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[67] Habib, M. F., Tornatore, M., Dikbiyik, F., and Mukherjee, B. Disaster survivability in optical communication networks. *Computer Communications 36*, 6 (2013), 630–644.

[68] Hansen, A., Kvalbein, A., Cicic, T., and Gjessing, S. Resilient routing layers for network disaster planning. In *Networking - ICN 2005*, P. Lorenz and P. Dini, Eds., vol. 3421 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005, pp. 1097–1105.

[69] Haverkort, B. R. *Performance of Computer Communication Systems: A Model-Based Approach*. John Wiley & Sons, Inc., 1998.

[70] Heindl, A., Mitchell, K., and van de Liefvoort, A. Correlation bounds for second-order {MAPs} with application to queueing network decomposition. *Performance Evaluation 63*, 6 (2006), 553 – 577. Modelling Techniques and Tools for Computer Performance Evaluation.

[71] Hilkevitch, J. FAA outlines strategy to recover from air-traffic control outages. *Chicago Tribune* (2015). [Online; accessed 6-April-2016].

[72] Hirschman, D. 'ATC Zero': Inside the Chicago center fire. *Aircraft Owner and Pilots Association* (2014). [Online; accessed 6-April-2016].

[73] Horváth, A., Horváth, G., and Telek, M. A joint moments based analysis of networks of MAP/MAP/1 queues. *Performance Evaluation 67*, 9 (2010), 759–778.

[74] Horváth, G., and Telek, M. Acceptance-rejection methods for generating random variates from matrix exponential distributions and rational arrival processes. In *Matrix-Analytic Methods in Stochastic Models*. Springer, 2013, pp. 123–143.

[75] Jackson, J. R. Networks of waiting lines. *Operations Research 5*, 4 (1957), 518–521.

[76] Jarvis, J. P., and Shier, D. R. An improved algorithm for approximating the performance of stochastic flow networks. *INFORMS Journal on Computing 8*, 4 (1996), 355–360.

[77] Kim, S. The two-moment three-parameter decomposition approximation of queueing networks with exponential residual renewal processes. *Queueing Systems 68*, 2 (2011), 193–216.

[78] Kim, S., Muralidharan, R., and O'Cinneide, C. A. Taking account of correlations between streams in queueing network approximations. *Queueing Systems 49*, 3-4 (2005), 261–281.

[79] Kounev, S. Performance modeling and evaluation of distributed component-based systems using queueing petri nets. *Software Engineering, IEEE Transactions on 32*, 7 (2006), 486–502.

[80] Kramer, W., and Langenbach-Belz, M. Approximate formulae for the delay in the queueing system GI/G/1. In *Proceedings ITC* (1976), vol. 8, pp. 235–1.

[81] Kumaran, J., Mitchell, K., and van de Liefvoort, A. Characterization of the departure process from an ME/ME/1 queue. *RAIRO-Operations Research 38*, 02 (2004), 173–191.

[82] Kumaran, J., Mitchell, K., and van de Liefvoort, A. A spectral approach to compute performance measures in a correlated single server queue. *SIGMETRICS Perform. Eval. Rev. 33*, 2 (Sept. 2005), 12–14.

[83] Kuperman, G., Modiano, E., and Narula-Tam, A. Analysis and algorithms for partial protection in mesh networks. *Journal of Optical Communications and Networking 6*, 8 (2014), 730–742.

[84] K.U. ResiliNets Group. ResiliNets Topology - Map Viewer. http://www.ittc.ku.edu/resilinets/maps/, 2012. [Online; accessed 30-October-2013].

[85] Kvalbein, A., Hansen, A., Cicic, T., Gjessing, S., and Lysne, O. Fast IP network recovery using multiple routing configurations. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings* (2006), pp. 1–11.

[86] Kvalbein, A., Hansen, A. F., Čičic, T., Gjessing, S., and Lysne, O. Multiple routing configurations for fast IP network recovery. *IEEE/ACM Transactions on Networking (TON) 17*, 2 (2009), 473–486.

[87] Lee, H.-W., Modiano, E., and Lee, K. Diverse routing in networks with probabilistic failures. *Networking, IEEE/ACM Transactions on 18*, 6 (2010), 1895–1907.

[88] Lee, K., Modiano, E., and Lee, H.-W. Cross-layer survivability in WDM-based networks. *IEEE/ACM Transactions on Networking (TON) 19*, 4 (2011), 1000–1013.

[89] Li, R., Wang, X., and Jiang, X. Network survivability against region failure. In *Signal Processing, Communications and Computing (ICSPCC), 2011 IEEE International Conference on* (2011), IEEE, pp. 1–6.

[90] Li, V. O., and Silvester, J. A. Performance analysis of networks with unreliable components. *Communications, IEEE Transactions on 32*, 10 (1984), 1105–1110.

[91] Lipsky, L. R. *Queueing Theory: A Linear Algebraic Approach*. Springer, 2009.

[92] Long, X., Tipper, D., and Gomes, T. Measuring the survivability of networks to geographic correlated failures. *Optical Switching and Networking 14* (2014), 117–133.

[93] Madory, D. Renesys Blog, Hurricane Sandy: Global impact. http://www.renesys.com/2012/11/sandys-global-impacts/, 2012. [Online; accessed 30-October-2013].

[94] Manzano, M., Calle, E., Torres-Padrosa, V., Segovia, J., and Harle, D. Endurance: A new robustness measure for complex networks under multiple failure scenarios. *Computer Networks 57*, 17 (2013), 3641–3653.

[95] Marshall, K. T. Some inequalities in queuing. *Operations Research 16*, 3 (1968), 651–668.

[96] MathWorks, I. *MATLAB: the language of technical computing. Desktop tools and development environment, version 7*, vol. 9. MathWorks, 2005.

[97] McQuillan, J., Richer, I., and Rosen, E. The new routing algorithm for the ARPANET. *Communications, IEEE Transactions on 28*, 5 (1980), 711–719.

[98] Medhi, D. Diverse routing for survivability in a fiber-based sparse network. In *Proc. of IEEE International Conference on Communications (ICC'1991)* (Denver, CO, June 1991), pp. 672–676.

[99] Medhi, D. A Unified Approach to Network Survivability for Teletraffic Networks: Models, Algorithms and Analysis. *IEEE Trans. on Communications 42* (1994), 534–548.

[100] Medhi, D. A unified approach to network survivability for teletraffic networks: Models, algorithms and analysis. *Communications, IEEE Transactions on 42*, 234 (1994), 534–548.

[101] Medhi, D. Network Reliability and Fault-Tolerance. *Wiley Encyclopedia of Electrical and Electronics Engineering* (1999).

[102] Medhi, D., and Khurana, R. Optimization and performance of network restoration schemes for wide-area teletraffic networks. *Journal of Network and Systems Management 3* (1995), 265–294.

[103] Medhi, D., and Khurana, R. Optimization and performance of restoration schemes for wide-area teletraffic networks. *Journal of Network and Systems Management 3*, 3 (1995), 265–294.

[104] Menasce, D. QoS issues in web services. *Internet Computing, IEEE 6*, 6 (2002), 72–75.

[105] Menascé, D. A., and Almeida, V. *Capacity Planning for Web Services: Metrics, Models, and Methods*. Prentice Hall PTR, 2001.

[106] Menth, M., Duelli, M., Martin, R., and Milbrandt, J. Resilience analysis of packet-switched communication networks. *IEEE/ACM Transactions on Networking (TON) 17*, 6 (2009), 1950–1963.

[107] Menth, M., Hartmann, M., and Martin, R. Robust IP link costs for multilayer resilience. In *NETWORKING 2007. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*. Springer, 2007, pp. 749–761.

[108] Menth, M., and Martin, R. Network resilience through multi-topology routing. In *The 5th International Workshop on Design of Reliable Communication Networks* (2005), pp. 271–277.

[109] Meyer, J. F. On evaluating the performability of degradable computing systems. *Computers, IEEE Transactions on 100*, 8 (1980), 720–731.

[110] Mitchell, K., Sohraby, K., van de Liefvoort, A., and Place, J. Approximation models of wireless cellular networks using moment matching. *Selected Areas in Communications, IEEE Journal on 19*, 11 (2001), 2177–2190.

[111] Mitchell, K., and van de Liefvoort, A. Approximation models of feed-forward GG1N queueing networks with correlated arrivals. *Performance Evaluation 51*, 2 (2003), 137–152.

[112] Molisch, A. F. *Wireless Communications*, vol. 15. John Wiley & Sons, 2010.

[113] National Aeronautics and Space Administration (NASA). NASA, Virtual Skies, Air Traffic Management System. http://virtualskies.arc.nasa.gov/atm/index.html. [Online; accessed 6-April-2016].

[114] Neumayer, S., and Modiano, E. Network reliability with geographically correlated failures. In *INFOCOM, 2010 Proceedings IEEE* (2010), IEEE, pp. 1–9.

[115] Octave community. GNU/Octave, 2012.

[116] Oikonomou, K. N., Sinha, R. K., and Doverspike, R. D. Multi-layer network performance and reliability analysis. *International Journal of Interdisciplinary Telecommunications and Networking (IJITN) 1*, 3 (2009), 1–30.

[117] Oki, E., Matsuura, N., Shiomoto, K., and Yamanaka, N. A disjoint path selection scheme with shared risk link groups in GMPLS networks. *Communications Letters, IEEE 6*, 9 (2002), 406–408.

[118] OPNET Technologies. http://www.opnet.com, 2010.

[119] Orlowski, S., Wessäly, R., Pióro, M., and Tomaszewski, A. SNDlib 1.0-Survivable network design library. *Networks 55*, 3 (2010), 276–286.

[120] Pacharintanakul, P., and Tipper, D. Crosslayer survivable mapping in Overlay-IP-WDM networks. In *Design of Reliable Communication Networks, 2009. DRCN 2009. 7th International Workshop on* (2009), IEEE, pp. 168–174.

[121] Pióro, M., and Medhi, D. *Routing, flow, and capacity design in communication and computer networks*. Elsevier, 2004.

[122] Pongthawornkamol, T., Nahrstedt, K., and Wang, G. Probabilistic QoS modeling for reliability/timeliness prediction in distributed content-based publish/subscribe systems over best-effort networks. In *Proceedings of the 7th international conference on Autonomic computing* (2010), ACM, pp. 185–194.

[123] Principe, J. C., Euliano, N. R., and Lefebvre, W. C. *Neural and adaptive systems: fundamentals through simulations with CD-ROM*. John Wiley & Sons, Inc., 1999.

[124] Przygienda, T., Shen, N., and Sheth, N. M-ISIS: Multi topology (MT) routing in intermediate system to intermediate systems (IS-ISs). RFC 5120, RFC Editor, February 2008.

[125] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and Pillay-Esnault, P. Multi-topology (MT) routing in OSPF. RFC 4915, RFC Editor, June 2007.

[126] Rabta, B. A hybrid method for performance analysis of G/G/m queueing networks. *Mathematics and Computers in Simulation 89* (2013), 38–49.

[127] Rahman, M. R., and Boutaba, R. SVNE: Survivable virtual network embedding algorithms for network virtualization. *Network and Service Management, IEEE Transactions on 10*, 2 (2013), 105–118.

[128] Rahnamay-Naeini, M., Pezoa, J. E., Azar, G., Ghani, N., and Hayat, M. M. Modeling stochastic correlated failures and their effects on network reliability. In *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on* (2011), IEEE, pp. 1–6.

[129] Reinecke, P., and Horváth, G. phase-type distributions for realistic modelling in discrete-event simulation. In *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques* (2012), ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 283–290.

[130] Ridout, M. Generating random numbers from a distribution specified by its laplace transform. *Statistics and Computing 19*, 4 (2009), 439–450.

[131] Rissanen, J. Hypothesis selection and testing by the MDL principle. *The Computer Journal 42*, 4 (1999), 260–269.

[132] Robertson, G., and Nelakuditi, S. Handling multiple failures in IP networks through localized on-demand link state routing. *Network and Service Management, IEEE Transactions on 9*, 3 (2012), 293–305.

[133] Rohrer, J. P., Jabbar, A., and Sterbenz, J. P. Path diversification for future internet end-to-end resilience and survivability. *Telecommunication Systems 56*, 1 (2014), 49–67.

[134] S. Bryant, S. Previdi, M. S. IP fast reroute using not-via addresses. Internet-Draft draft-ietf-rtgwg-ipfrr-notvia-addresses-09.txt, IETF Secretariat, Feb. 2011.

[135] Sachs, K., Kounev, S., Bacon, J., and Buchmann, A. Performance Evaluation of Message-Oriented Middleware Using the SPECjms2007 Benchmark. *Performance Evaluation 66*, 8 (2009), 410–434.

[136] Sadre, R., and Haverkort, B. R. Decomposition-based queueing network analysis with FiFiQueues. In *Queueing Networks*. Springer, 2011, pp. 643–699.

[137] Saito, H. Analysis of geometric disaster evaluation model for physical networks. *Networking, IEEE/ACM Transactions on 23*, 6 (2015), 1777–1789.

[138] Scheffel, M. C., Gruber, C. G., Schwabe, T., and Prinz, R. G. Optimal multi-topology routing for IP resilience. *AEU - International Journal of Electronics and Communications 60*, 1 (2006), 35 – 39.

[139] Schwarz, G., et al. Estimating the dimension of a model. *The annals of statistics 6*, 2 (1978), 461–464.

[140] Sen, A., Murthy, S., and Banerjee, S. Region-based connectivity-a new paradigm for design of fault-tolerant networks. In *High Performance Switching and Routing, 2009. HPSR 2009. International Conference on* (2009), IEEE, pp. 1–7.

[141] Shand, M., and Bryant, S. IP fast reroute framework. RFC 5714, RFC Editor, January 2010.

[142] Srivastava, S., Thirumalasetty, S. R., and Medhi, D. Network traffic engineering with varied levels of protection in the next generation Internet. In *Performance Evaluation and Planning Methods for the Next Generation Internet*. Springer, 2005, pp. 99–124.

[143] Standley, J., Brown, V., Comitz, P., and Schoolfield, J. SWIM segment 2 deployment and utilization in NextGen R&D Programs. In *Integrated Communications, Navigation and Surveillance Conference (ICNS), 2012* (2012), pp. G8–1–G8–5.

[144] Stephan, K. D. We've got to talk: Emergency communications and engineering ethics. In *Technology and Society, 2006. ISTAS 2006. IEEE International Symposium on* (2006), IEEE, pp. 1–7.

[145] Sterbenz, J. P., Cetinkaya, E. K., Hameed, M. A., Jabbar, A., Qian, S., and Rohrer, J. P. Evaluation of network resilience, survivability, and disruption tolerance: analysis, topology generation, simulation, and experimentation. *Telecommunication Systems* (2011), 1–32.

[146] Sterbenz, J. P., Cetinkaya, E. K., Hameed, M. A., Jabbar, A., and Rohrer, J. P. Modelling and analysis of network resilience. In *Communication Systems and Networks (COMSNETS), 2011 Third International Conference on* (2011), IEEE, pp. 1–10.

[147] Sterbenz, J. P., Hutchison, D., Çetinkaya, E. K., Jabbar, A., Rohrer, J. P., Schöller, M., and Smith, P. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer Networks 54*, 8 (June 2010), 1245–1265.

[148] Telek, M., and Horváth, G. A minimal representation of markov arrival processes and a moments matching method. *Performance Evaluation 64*, 9 (2007), 1153–1168.

[149] University of Missouri Kansas City. An analysis of the FAA NextGen SWIM architecture. FAA grant proposal, Federal Aviation Administration, February 2011.

[150] U.S. Senate Committee on Commerce Science and Transportation. Congestion and delays: The impact on passengers and possible solutions. *U.S. Government Printing Office* (2007). [Online; accessed 6-April-2016].

[151] Van de Liefvoort, A. The moment problem for continuous distributions. *Technical Report, University of Missouri, WP-CM-1990-02, Kansas City* (1990).

[152] van De Liefvoort, A. The waiting-time distribution and its moments of the PH/PH/1 queue. *Operations Research Letters 9*, 4 (1990), 261–269.

[153] Vasseur, J.-P., Pickavet, M., and Demeester, P. *Network recovery: Protection and restoration of optical, SONET-SDH, IP, and MPLS*. Elsevier, 2004.

[154] Whitt, W. Performance of the queueing network analyzer. *Bell System Technical Journal, The 62*, 9 (1983), 2817–2843.

[155] Whitt, W. The queueing network analyzer. *Bell System Technical Journal 62*, 9 (1983), 2779–2815.

[156] Wilkinson, J. H. *The Algebraic Eigenvalue Problem*, vol. 155. Oxford Univ Press, 1965.

[157] Wong, J., Robinson, C., et al. Urban search and rescue technology needs: identification of needs. *Federal Emergency Management Agency (FEMA) and the National Institute of Justice (NIJ). Document*, 207771 (2004).

[158] Yu, H., Qiao, C., Anand, V., Liu, X., Di, H., and Sun, G. Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE* (Dec 2010), pp. 1–6.

[159] Zhang, D., and Sterbenz, J. P. G. Modelling Critical Node Attacks in MANETs. In *Self-Organizing Systems: 7th IFIP TC 6 International Workshop, IWSOS 2013, Palma de Mallorca, Spain, May 9-10, 2013, Revised Selected Papers*, W. Elmenreich, F. Dressler, and V. Loreto, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 127–138.

VITA

Michael (Todd) Gardner was born on September 16, 1968 in Joplin, Missouri where he graduated from Joplin High School in 1986. He went on to receive a Bachalor of Science in Electrical Engineering and Computer Engineering from the University of Missouri, Columbia in 1990. In 2002, he received his Master of Science in Electrical Engineering from the University of Kansas.

Todd has worked over 20 years for the U.S. Federal Aviation Administration (FAA) on air traffic control systems as a communications and networking engineer. He has served as the lead engineer on several notable projects including the FAA's Time Division Multiplex to Internet Protocol (TDM-to-IP) effort and the FAA Telecommunications Infrastracture 2 (FTI-2) acquisition.

His research interests include high availability networks, resilient networks, and performance analysis with journal and conference publications related to these areas of research. He was honored with best paper awards at IEEE CQR in 2011, IEEE DRCN in 2014, and runner up for best paper at IEEE DRCN 2015. These were all for research work related to resilient networks. In 2015, the UMKC CSEE Department presented him with a CSEE Outstanding Ph.D. student award.

Todd is also a Registered Professional Engineer in the State of Missouri and an IEEE and HKN member.