Future Internet Routing Design for Massive Failures and Attacks

By

Yufei Cheng

Submitted to the graduate degree program in Electrical Engineering & Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Chairperson: Prof. James P.G. Sterbenz

Prof. Jiannong Cao

Prof. Victor S. Frost

Prof. Fengjun Li

Prof. Deep Medhi

Prof. Gary J. Minden

Prof. Michael S. Vitevitch

Date Defended: 6-June-2016

The Dissertation Committee for Yufei Cheng certifies that this is the approved version of the following dissertation:

Future Internet Routing Design for Massive Failures and Attacks

Chairperson: Prof. James P.G. Sterbenz

Date approved: 6-June-2016

Abstract

Given the high complexity and increasing traffic load of the Internet, geo-correlated challenges caused by large-scale disasters or malicious attacks pose a significant threat to dependable network communications. To understand its characteristics, we propose a critical-region identification mechanism and incorporate its result into a new graph resilience metric, compensated Total Geographical Graph Diversity. Our metric is capable of characterizing and differentiating resiliency levels for different physical topologies. We further analyze the mechanisms attackers could exploit to maximize the damage and demonstrate the effectiveness of a network restoration plan. Based on the geodiversity in topologies, we present the *path geodiverse problem* and two heuristics to solve it more efficiently compared to the optimal algorithm. We propose the *flow geodiverse problem* and two optimization formulations to study the tradeoff among cost, end-to-end delay, and path skew with multipath forwarding. We further integrate the solution to above models into our cross-layer resilient protocol stack, ResTP–GeoDivRP. Our protocol stack is prototyped and implemented in the network simulator *ns-3* and emulated in our KanREN testbed. By providing multiple GeoPaths, our protocol stack provides better path restoration performance than Multipath TCP.

Page left intentionally blank.

Contents

Page left intentionally blank.

List of Figures

Page left intentionally blank.

List of Tables

Page left intentionally blank.

Chapter 1

Introduction and Motivation

The demands for Internet resilience have been increasing tremendously and it is important to analyze their resilience to various faults and challenges [3]. Networks are generally studied as pure graphs without considering the geographical properties of nodes and links [4]. Random link/node and non-correlated failures are widely studied for IP networks [5–7]. Multiple Routing Configurations (MRC) in IP networks [8] have been proposed to guarantee the recovery of a single link failure and have been shown to be effective and scalable. For dual-link failures, a fast recovery mechanism has been proposed [9]. However, most of these works focus on random challenges using either random synthetic or IP-layer topologies. Mechanisms have been proposed to identify link- and PoP-disjoint paths for the same node pair in Internet service providers (ISPs) networks to improve network resilience [10]. Several attack techniques based on random failures are presented for the IP networks [5, 6], and an IP-level restoration [11] mechanism has been shown to be effective.

Network components in the physical adjacency may fail together during a natural disaster such as an electrical blackout or an earthquake, or malicious attacks; these are the geo-correlated challenges and can result in significant damage to dependable network communications [3]. The impact on the Internet from geo-correlated challenges is still an

open issue. When the same intensity of challenges occur at different physical locations, the damage to the network connectivity varies greatly. Detection of the vulnerable areas or critical regions has several practical applications, fibers in these regions can be either protected by shielding, strengthening, or closely monitoring for resilient network communication. Local graph metrics such as centrality metrics have been used in network vulnerability analysis [12,13]. We employ centrality metrics to guide the selection among the failed nodes for prioritized protection in the face of regional challenges. We present the performance improvement from the prioritized protection through graph analysis and further verify our graph analysis using network simulations. As far as we know, this is the first work to use centrality metrics in prioritizing the restoration of network services during regional challenges.

Since the regional challenge effect is frequently long-term [14], a set of backup paths is required for survivable routing. The single-location challenge scenario has been analyzed and a polynomial algorithm has been formulated [15]. Correlated and simultaneous challenges have been discussed [16], and different circular-shaped vulnerability points have been identified [17]. The vulnerability analysis result can not only guide the network design, but also help design resilient network architecture to consider geodiverse paths. However, none of this work has focused on a reliable cross-layer network architecture to cope with large-scale regional challenges.

An Internet Service Provider (ISP) network is an entity that manages a set of nodes and links to provide Internet services. Each ISP has the full control of which intradomain routing protocol to run and it can be redesigned to consider multiple alternative paths. Traditional intradomain routing protocols, such as Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS), are designed to form a single shortest path for each node pair. Although this ensures simple loop-less routing operation, alternate path needs to be calculated if the primary one fails. This reconvergence

process is usually fairly slow due to the protocol's hold down timers [18].

In order to quickly bypass the failed region, a resilient protocol is required to quickly find a single or multiple alternative paths for the communicating node pairs. Some may argue that in order to increase network resilience to regional failures, it should be considered during the network planning phase; however, this is not always the case for the following reasons. First, network planning with over-provisioning is a long term process; the high cost and policy limitation of deploying new fibers have hindered the improvement of resilience through new physical-level diversity. From a network protocol level, if several geodiverse paths can be quickly calculated after the challenge, the resilience of the current network can be improved tremendously without new network components. Second, although sophisticated network planning mechanisms can help reduce the impact to network traffic during area-based challenges, resilient routing is still needed to get around challenged areas quickly and be adaptive to traffic and congestion in the network. This is the one of the main motivations behind this work.

We study the path geodiverse problem (PGD) and the flow geodiverse problem (FGD) in this work. The motivation is that most networked devices have access to multiple partial or complete physical-layer paths between endpoints, and many of these paths have a certain degree of geographic diversity. However, we are currently unable to benefit from it since design decisions in the current Internet protocol stack assume unipath and shortest path routing. This dramatically decreases the ability to provide resilience under either targeted attacks or large-scale disasters. We can achieve improved performance and increased resilience with multiple geodiverse paths.

1.1 Thesis Statement

To understand the resilience of physical networks under geo-correlated challenges, critical regions are required to be identified before further design decisions can be made. Moreover, a graph metric is required to quantify the relative resilience of different physical networks with the identified critical regions. Finally, routing and transport protocol redesign is required to improve the end-to-end resilience under large-scale disasters or malicious attacks.

Therefore, our thesis statement is:

> *Network resilience against regional disasters requires analysis and fully understanding its characteristics. A critical region identification mechanism can help network operators to concentrate monitoring and protection of resources in these areas. Geodiverse routing protocol design can improve overall network resilience under regional challenges.*

1.2 Proposed Solution

We propose a critical-region identification mechanism using a moving-circle challenge model [19]. Other models, such as scaling-circle and polygon challenges [3, 20] are similarly applicable and we plan to include such analysis in our future work. This mechanism captures the essence of physical challenges while maintaining simplicity and effectiveness. Based on the identified critical regions, we extend the path-diversification metric to consider the *geographic separation* of nodes and links for resiliency analysis. This is an extension to our previous mechanism [21, 22] in order to represent graph resilience to geo-correlated challenges, as opposed to only individual node or link outages. We present our GeoPath diversity metric: minimum distance d between any two nodes on alternate paths.

Based on the geodiversity of different node pairs, we present *path geodiversitification* – a new mechanism (proposed in [21,22]) to quantify the graph GeoPath diversity by selecting multiple geodiverse paths between a given node pair to achieve high network survivability. We apply this mechanism in several Internet service providers (ISPs) optical-fiber networks to compare their relative robustness against regional challenges. This mechanism allows future internetworking architectures to exploit naturally rich physical topologies to a far greater extent than is possible with only shortest-path routing or equal-cost load balancing.

Furthermore, we formulate the path geodiverse problem (PGD) and propose heuristics to efficiently solve it. The solution is incorporated in the GeoPath Diverse Routing Protocol (GeoDivRP), which provides multiple geodiverse paths to circumvent regional failures given a threat model. We integrate GeoDivRP into our ResTP–GeoDivRP protocol stack for resilient network communications. *Knobs* and *dials* are used between GeoDivRP and ResTP for cross-layer communication. We apply our multipath algorithm in several real-world ISPs networks to analyze the diversity gain and improvement in packet delivery ratio (PDR) as well as average throughput.

To better allocate traffic for ResTP–GeoDivRP protocol stack, we formulate an optimization problem to minimize the delay and skew product among the multiple paths calculated for each node pair. The solution provides better link traffic utilization and throughput compared to Open Shortest Path First (OSPF). With the calculated bounded-skew geodiverse paths using the iWPSP heuristic for GeoDivRP [23], our protocol increases the throughput compared to OSPF under regional challenges. Past work has studied the bounded buffer problem but have assumed a maximum path-length constraint. Our heuristic does not restrict the maximum path length since it may lead to no usable skew-bounded paths. We argue that for physical topologies, it is not necessary to set an upper bound for path length as the network diameter (longest shortest path for

all node pairs) is small for a mesh-like topology [24]. With the optimized diverse paths, ResTP–GeoDivRP presents a better performance compared to the well-known Multipath TCP (MPTCP) protocol [25, 26].

We further incorporate our cross-layer design in the software-defined networking (SDN) domain. By taking advantage of the failure detection model implemented in SDN, GeoDivRP responds to network failures much faster. Coupled with the optimization model, it realizes the minimized delay-skew product when decoupling traffic onto multiple paths. We evaluate our framework using our resilient transport protocol as well as MPTCP in the face of geo-correlated challenges. We further demonstrate our Web framework to automate the OpenFlow experiments by programmatically importing network topologies and perform failure experiments using the user-provided challenge regions.

It is rarely feasible to conduct network experiments on a production ISP network, especially at a national scale. Network researchers resort to simulations or emulations to study their ideas and proposals. In this work, we use *ns-3* [27] simulation software and *Mininet* [28] to prototype and analyze our protocol. *ns-3* is a popular network simulator to analyze network protocols while *Mininet* is a network emulator that using Linux Kernel code to emulate network applications. Testbed experiments from our KanREN [29] network have been included as well. As for traffic optimization, we use the *OpenOpt* optimization framework [30] for solving the two optimization problems and use real-world network topologies from KU TopView [31, 32].

1.3 Contributions

The main contributions of this dissertation are as follows:

1. A critical-region identification mechanism using a moving circle challenge model.

This enables one to analyze a topology's geographic resilience for a given region.

2. A graph metric for quantifying regional challenges in physical networks. This enables one to compare the relative resilience among different topology.

3. Analyze the effectiveness of a network protection plan. With the plan in place, a network presents better resilience against challenges.

4. Implementation of our GeoPath Diverse Routing Protocol (GeoDivRP) routing algorithm in *ns-3* network simulator with the integration with ResTP to form a resilient network protocol stack, ResTP–GeoDivRP. This enables either alternative path for fail-over or multipath diverse paths for improved throughput. In the performance comparison against MPTCP, our protocol stack performs better.

5. Development of GeoDivRP in software-defined networking (SDN) domain and use an OpenFlow controller to control the routing and optimization mechanism.

6. Emulation in Mininet as well as experiments using physical OpenFlow switches deployed in the KanREN [29] testbed.

1.4 List of Related Publications

1.4.1 Journal Papers

1. **Yufei Cheng**, Deep Medhi, and James Sterbenz. Geodiverse Routing with Path Delay and Skew Requirement under Area-based Challenges Networks journal (Wiley), Volume 66, Issue 4, pp. 335–346, Dec. 2015.

2. **Yufei Cheng**, M. Todd Gardner, Junyan Li, Rebecca May, Deep Medhi, and James Sterbenz. Analyzing GeoPath Diversity and Improving Routing Performance in Optical Networks. Computer Networks, Volume 82, Issue C, pp. 50–67, May 2015.

3. Egemen K. Cetinkaya, Mohammed J.F. Alenazi, **Yufei Cheng**, Andrew M. Peck, and James P.G. Sterbenz. A comparative analysis of geometric graph models for modeling backbone networks. Optical Switching and Networking, Volume 14, Part 2, pp. 95–106, 2014.

1.4.2 Peer-Reviewed Proceedings

1. **Yufei Cheng**, Truc Anh Ngoc Nguyen, Md. Moshfequr Rahman, Siddharth Gangadhar, Mohammed J.F. Alenazi, and James P.G. Sterbenz. Cross-Layer Geodiverse Protocol Stack for Resilient Multipath Transport and Routing using OpenFlow. 12th International Conference on Design of Reliable Communication Networks, Paris, France, March 2016, pp. 103–105.

2. **Yufei Cheng**, Md. Moshfequr Rahman, Siddharth Gangadhar, Mohammed J.F. Alenazi, and James P. G. Sterbenz. Cross-Layer Framework with Geodiverse Routing in Software-Defined Networking. 2nd International Workshop on Management of SDN and NFV Systems, Barcelona, Spain, November 2015, pp. 348–353.

3. **Yufei Cheng** and James P. G. Sterbenz. Critical Region Identification and Geodiverse Routing Protocol under Massive Challenges. Proceedings of the 7th IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM), Munich, Germany, October 2015, pp. 14–20,

4. **Yufei Cheng**, M. Todd Gardner, Junyan Li, Rebecca May, Deep Medhi, and James P.G. Sterbenz Optimized Heuristics for a Geodiverse Routing Protocol. Proceedings of the IEEE 10th International Workshop on the Design of Reliable Communication Networks (DRCN), Ghent, Belgium, 2014, pp. 1–9.

5. **Yufei Cheng** and James P. G. Sterbenz. GeoDivRP Routing with Path Jitter Requirement under Regional Challenges. Proceedings of the 6th IEEE/IFIP Interna-

tional Workshop on Reliable Networks Design and Modeling (RNDM), Barcelona, Spain, November 2014, pp. 179—186.

6. **Yufei Cheng**, Junyan Li, and James P. G. Sterbenz Path Geo-diversification: Design and Analysis. Proceedings of the 5th IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM), Almaty, 2013, pp. 50–67.

7. Egemen K. Çetinkaya, Mohammed J. F. Alenazi, **Yufei Cheng**, Andrew M. Peck, and James P. G. Sterbenz On the Fitness of Geographic Graph Generators for Modelling Physical Level Topologies. Proceedings of the 5th IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM), Almaty, 2013, pp. 38–45.

1.4.3 Additional Publications

1. Mohammed J. F. Alenazi, **Yufei Cheng**, Dongsheng Zhang, and James P. G. Sterbenz. Epidemic routing protocol implementation in ns-3. Proceedings of the 2015 Workshop on ns-3 (WNS3 '15). ACM, New York, NY, USA, 2015, pp. 83–90.

1.5 Organization

In this chapter, we provide an overview and motivation for this dissertation. The remainder of this work is organized as follows: Chapter 2 presents the background and related work. Chapter 3 describes the geo-correlated challenges and introduce our critical region identification mechanism; we further analyze malicious attacks and provide restoration suggestions. Chapter 4 presents the path geodiverse problem (PGD) and our resilient routing protocol GeoDivRP to solve the problem. We further our discussion by providing an overview of the ResTP–GeoDivRP framework. In Chapter 5, we present two optimization formulations for the flow geodiverse problem (FGD). The controlled path delay

and skew product optimization for GeoDivRP has been proposed; we further introduce the implementation of ResTP–GeoDivRP in *ns-3*, and provide the performance evaluation using either single-path and multipath forwarding. Chapter 6 presents our work using software-defined networking (SDN). Using our Web framework, we present initial experiment results comparing GeoDivRP to OSPF in the face of regional failures.

Chapter 2

Background and Related Work

In this chapter, we present the background and related work of this dissertation. First, we introduce geo-correlated challenges and its significant threat to network resilient in Section 2.1. An overview of diversity in network topology and definition of geographic diversity are presented in Section 2.2. Existing geodiverse protocols are discussed in Section 2.3. An overview of network optimization is presented in Section 2.4. Finally, Section 2.5 discusses the software-defined networking.

2.1 Geo-Correlated Challenge

A *geo-correlated challenge* is defined as a failure or malicious attack that affects a set of nodes and links in a geographic vicinity. It has been observed that geo-correlated challenges can cause a large number of failures in a geographic region and give rise to significant damage to network communications [33,34]. Computer networks are susceptible to this type of challenge from natural disasters such as: earthquakes, tornados, solar flares, floods, and malicious attacks [35–38]. Natural disasters such as earthquakes can cause catastrophic damage to network communications. For example, the 2015 Nepal Earthquake affected Kathmandu, 80 km (50 mi) from the epicenter as shown in Fig-

ure 2.1. Another example is the two severe power blackouts affected most of northern and eastern India on 30 and 31 July 2012. As shown in Figure 2.2, darker red color shows the states that were down for two days and the lighter red down for one day.

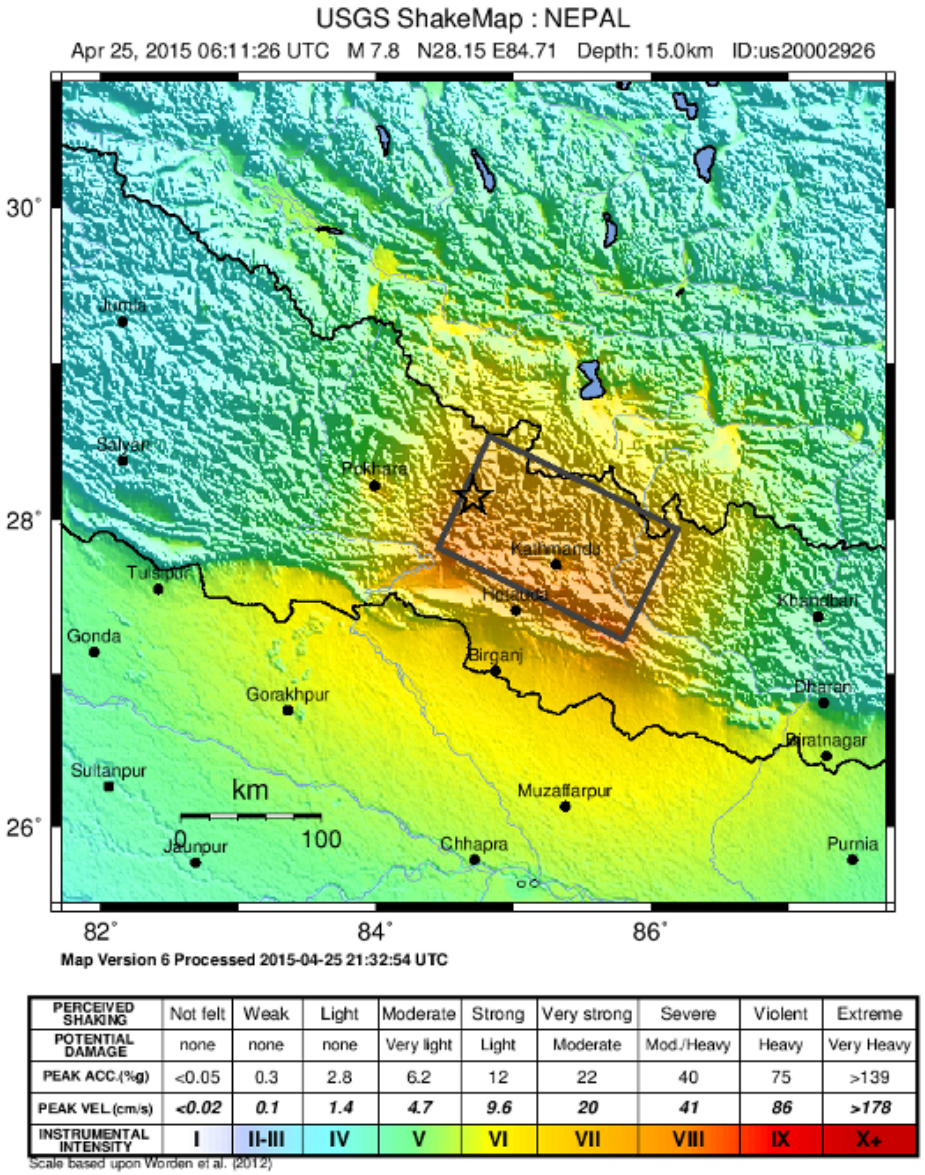

Figure 2.1: 2015 Nepal earthquake USGS shakeMap [1]

Figure 2.2: 2012 Indian blackout affected states [2]

2.1.1 Malicious Attacks and Centrality Metrics

Malicious attacks are caused by attackers targeted at the networks with malevolent intention [39]. For example, a denial-of-service (DoS) attack is an attempt to make network resources unavailable to legitimate users by crowding it with a large amount of traffic. An electromagnetic pulse (EMP) attack can disrupt electronic devices in a large range. If targeted at a vulnerable region, the damage is even more significant.

Centrality metrics have been used to study the performance of networks against malicious attacks [19, 40–42]. In network science and graph theory, centrality measures indicate the importance of a node in a graph and more damage occurs with its removal. Centrality-balanced improvement mechanism has been proposed to guide the network design using centrality metrics [43, 44]. From the attackers' perspective, they may study the network characteristics with a malicious intent to maximize the attack damage. An attack module [41] starts by sorting the different centrality measures from the highest to the lowest, and the node with the highest is failed from the graph in each iteration.

To study how attackers may exploit centrality measures to aid their malicious attack, we include degree, betweenness, closeness, eigenvector, and load centrality [45,46] metrics in our study. Degree centrality was the first and simplest centrality measure; it is the number of links affiliated to a node and is usually viewed as the relative importance of a node [47]. There are usually two degree centralities, indegree and outdegree. The indegree is the number of links connecting to this node while the outdegree is the number of links going out. Betweenness centrality is defined as the number of the shortest paths that pass through a node. It signifies a node's importance in the network communication [48] by qualifying the number of times a node acts as the bridge to other nodes. Closeness is the inverse of the sum of the shortest paths from a node to every other node and indicates the efficiency for spreading a packet in a network [47]. Therefore, based on closeness centrality, the more central a node is, the closer it is to all the other nodes. Eigenvector centrality is a another centrality measure [49] and a score is assigned to all nodes assuming that links to high-score nodes contribute more than links to low-score ones. The load centrality of a node is the fraction of load through that node compared to all the load in the network [48]. Each node sends a unit flow along the shortest path to every other node and the total amount of flow for each node is defined as the load.

2.1.2 Real-World Topology Data

For accurate analysis of the network resilience, it is important to use real-world topologies or structurally similar ones. Several projects have dedicated to network topology data storage, representation, and analysis. Rocketfuel [50] is a tool to collect, measure, and analyze IP-level topologies. Past work [51] has analyzed the characteristics and implications of the infrastructure sharing in the US longhaul fiber-optic network. SNDlib [52] is a library for the survivable telecommunication network design and several physical networks with link traffic capacities are provided. Furthermore, the project provides the benchmark result for several traffic optimization problems and offers guidance for the telecommunication network design. The Internet Topology Zoo [53] is an ongoing project to collect network topologies globally and they have both IP and Internet service providers (ISPs) fiber topologies. In this work, we study the topologies for Europe and South America in Chapter 3.

Most of the topologies used in this work are from the KU-TopView network topology viewer [31, 54] implemented by our ResiliNets [55] group for easier network graph visualization and storage. It is developed using the Google Map API and JavaScript to visually present the topological maps. It also provides multiple topology manipulation functions, such as merging different topologies into one and outputing its adjacency matrix. Furthermore, it can generate different synthetic topologies such as Gabriel graph [56, 57] and Waxman graph [58].

2.2 Diversity in Network Topology

In this section, we discuss the past work exploring the path diversity and the geographic diversity. Most networked devices have access to multiple partial or complete physical-

layer paths between endpoints, and many of these paths have a certain degree of diversity. However, we are currently unable to benefit from it since design decisions in the current Internet protocol stack assume unipath and the shortest path routing. This dramatically decreases the ability for the network to provide resilience under either large-scale natural disasters or malicious attacks.

2.2.1 Path Diversity

Path diversity has been studied from a topological perspective [59–61], in terms of multipath in routing layer [62–69], as well as multipath in transport layer [22,70]. The mechanisms to take advantage of a network's path diversity is a major research topic [21,22,63] and a path diversity mechanism for qualifying the network resilience has been proposed [22]. Before giving the definition of the path diversity, we start with the definition of a Path.

A network is represented by a connected directed graph $G(V, E)$, where V is the set of nodes (vertices) and E is the set of links (edges).

Path is defined as a vector that contains all links (edges) E and intermediate nodes (vertices) V from a source node to a destination node

$$p = V \cup E \tag{2.1}$$

We represent a path as the sequence of nodes $p = (v_0, v_1, ...v_h)$, such that, for $0 \leq n \leq h-1, (v_n, v_{n+1}) \in E$. Each path p has an associated cost $c(p)$ which denotes its cost per unit flow. Each link is associated with a capacity u_e that denotes the maximum amount of flow it can carry and a lower bound l_e that denotes the minimum amount. For most of the cases, the lower bound is zero.

For the shortest path p_0, the path diversity $D(p_k)$ for any other path p_k between the same source and destination is shown in Equation 2.2 [22, 63].

$$D(p_k) = 1 - \frac{|p_k \cap p_0|}{|p_0|} \tag{2.2}$$

As shown in Figure 2.3, assuming node 0 is the source and node 2 the destination, there are four potential paths: $p_0 = [0, 1, 2]$, $p_1 = [0, 3, 2]$, $p_2 = [0, 1, 3, 2]$, and $p_3 = [0, 3, 1, 2]$. For the shortest path p_0, the path elements set is $\{(0, 1), 1, (1, 2)\}$. The tuples $(0, 1)$ and $(1, 2)$ represent the links and the element 1 is the node 1 in the path with the source and destination nodes excluded. The path elements set for p_1 is $\{(0, 3), 3, (3, 2)\}$. Using Equation 2.2, the path diversity $D(p_1) = 1 - \frac{0}{3} = 1$. Using the same equation, $D(p_2) = 1 - \frac{2}{3} = \frac{1}{3}$ and $D(p_3) = 1 - \frac{2}{3} = \frac{1}{3}$.

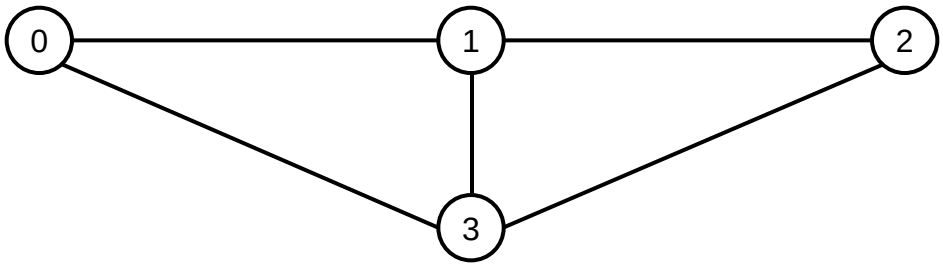

Figure 2.3: Path definition example

Path set denotes a set of paths between a node pair. For example, the path set for node 0 and 2 is $P_{0,2} = \{p_0, p_1, p_2, p_3\}$.

Path stretch is defined as the hop counts of H_{p_a} a given path p_a divided by the hop counts H_{p_s} of the shortest path p_s

$$h = H_{p_a}/H_{p_s}, \tag{2.3}$$

where we use the same definition from [63]. Note that in the protocol design presented in Chapter 4, we use path delay to represent path stretch.

Path skew is defined as the delay difference among multiple paths for a given node pair.

Given the definition of path diversity, a Shared Risk Link Group (SRLG) is defined as a set of links that share a common physical resource, such as cables or conducts, and can be affected simultaneously by a single failure or attack. SRLG-disjoint path is useful for dedicated path protection [71] and for addressing single or multiple physical challenges [72]. Physically Disjoint Paths (PDP) problem has been analyzed [73], and there are several studies [74–76] analyzing multiple failures in the context of SRLG. The survivable routing problem is shown to be NP-complete [77, 78] and an Integer Linear Programming (ILP) formulation has been proposed. Logical rings are used for protection against link failures and bound of number of links has been studied to improve the resilience for a given topology [79]. Path protection has been proposed to provide two SRLG-disjoint paths using graph transformation techniques [80]. Furthermore, a mechanism has been proposed for probabilistic correlated failure in SRLG [68].

2.2.2 Geographic Diversity

Geographic diversity [81] is proposed to represent how geographic separation the network components are, and it directly affects how a given network's resilience level is under geo-correlated challenges. It has been observed that the more dense a network is, the more vulnerable it is to regional challenges [82]. Geo-correlated challenges in the IP network have been considered and simulated [20, 83]. Since the challenge effect is frequently long-term [14], a set of backup paths are required for survivable routing. The single-location physical challenge scenario has been analyzed [84,85], as well as correlated

and simultaneous challenges [16]. Past work has studied the geographic vulnerabilities for several topologies [17]; based on the vulnerable areas identified, optimization mechanisms to alleviate these impacts have been proposed [86]. A random line-cut mechanism has been used to assess the vulnerability to regional-based challenges [15]. Region-based connectivity (RBC) has been proposed to analyze single and multiple failure region models [87, 88]. Wireless mesh network (WMN) survivability for regional failures has been analyzed [89] and a p-fractile region survivability function have been proposed. Both correlated failures and targeted attacks with simulation results have been presented [19].

2.2.3 Critical Region Identification

Critical node in a graph is a subset of nodes that with the removal, maximal damage is caused to the connectivity. A model to identify the critical infrastructures has been proposed [90]. The α-critical-distances mechanism has formulated a critical node identification mechanism with polynomial time complexity [33]. Several performance metrics such as the giant component size [91], average two terminal reliability [92] and network efficiency [93] are used. A probabilistic geo-correlated failure model has been analyzed [94] and the Strauss point process [95] has been used for either inhibition or clustering effects. Multiple attacks critical locations have been identified using computational geometry [15, 16]. Past work has proposed optimization mechanisms to reduce the searching complexity geographic vulnerabilities [17, 96].

A related notion to the critical node identification in the regionally correlated failure domain is the identification of critical regions. Several works have studied the geometric property of the network under regional challenges [97]. The smallest-enclosing circle problem [98] is used for critical region identification. Network vulnerability analysis has been done for multiple probabilistic physical attacks, and an approximation algorithm has been proposed to find the most vulnerable location [16, 99]. Critical region identification

models have been proposed for several failure shapes including circular, polygon, and ellipse [100].

Several events have demonstrated that geo-correlated challenges can be modeled as a moving circle with a given challenge radius. For example, an earthquake or hurricane normally has a failure radius from tens to hundreds of km [14, 39]. Other models, such as scaling-circle and polygon challenges [3, 20] are similarly applicable.

2.3 Geodiverse Protocols

Traditional intradomain routing protocols are designed to form a single shortest path for each source–destination pair due to its simplicity and efficiency. However, this comes with the cost of not having the option to choose an alternate path when the current one is unavailable due to failures or attacks. In order to quickly bypass the failed region, a resilient protocol is required to quickly find a single or multiple alternative paths for the communicating node pairs.

2.3.1 Current Intradomain Routing Protocol

Intradomain routing protocols run within an Internet Service Provider (ISP) network, and most ISPs run a link-state routing protocol based on configurable link weight. The link weights are tuned by network operators, such as for load balancing, failure avoidance, or security. The two primary intradomain routing protocols are Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS); we focus on OSPF in this work. Each router running OSPF has a static link weight configured for its outgoing link and the shortest path is calculated by Dijkstra's algorithm [101] based on the link-state advertisements (LSA) flooded throughout the network. Each router constructs its forwarding table based on the calculated shortest path. OSPF has several

advantages. First, routing is very simple since it is based on a single metric, the link weight. Second, by flooding the information in the network using the LSA packet, each router has a consistent view of the topology and routing loops are easily avoided. Finally, it is scalable using a hierarchy and reliable to single node failure due to its distributed nature.

However, there are some drawbacks due to its simple operation. First, the path is calculated only based on link weight, some other information, such as end-to-end delay, congestion level, cannot be considered during path calculation. Second, alternative paths are not provisioned and the path restoration process is slow. Although an end system or edge network has access to multiple paths, routers are not able to use the path diversity or the geodiversity the network topology has to offer [102, 103]. To control route flapping, OSPF introduces several timeout values, which slows the protocol convergence. Since OSPF needs to reconverge whenever the topology changes due to network failure, the process becomes even slower with geo-correlated challenges.

Several mechanisms have been proposed to address the above mentioned drawbacks. First, constrained shortest path first (CSPF) protocol is an extension to calculate the paths fulfilling a set of constraints, with different classes of traffic forwarded to different paths. However, the constraints cannot be dynamic demands such as path delay or jitter. In this work, we introduce our optimization engine to consider dynamic traffic information for optimal traffic distribution. Second, equal-cost multipath (ECMP) is commonly deployed where routers keep track of several shortest paths and then evenly split the flow among them [104]. Another mechanism for fast restoration is Fast-IP rerouting [105]. Instead of calculating one shortest path in the network restoration process, an alternative path is calculated to provide protection for the primary one.

2.3.2 Multipath Routing

Multiple shortest paths enable the network operators to balance load and provide better resilience by splitting traffic into multiple paths. Since each router has an updated view of the topology though LSAs, a k-shortest path algorithm can be used with k being the number of paths for each node pair. However, this is not realistic in practice due to its high computational cost. For example, for a single node pair, Yen's algorithm [106] has a complexity of $O(k|V|(m + |V| \log |V|))$ on a graph with $|V|$ nodes and $|E|$ links for a dense network. Furthermore, the forwarding table size will be k times larger. To reduce the complexity, path splicing [63] proposes a new routing mechanism that uses multiple instances of the link-state routing protocol; each link has a vector of link weights with each one tuned for different traffic class. For example, one can tune for high bandwidth use while another for low delay.

Several multipath architectures have been proposed for resilient traffic communication. A distributed traffic engineering heuristic, TeXCP, has been proposed that uses four paths for each demand [107], however, it may have potentially been misguided by a near-optimal solution. The cross-layer routing paths problem has been proposed by maximizing the objective function for users implementing multipath routing [108]. When regional failures occur, the rerouting traffic has the tendency to share common links in the vicinity of the threat zone and increase the congestion possibility. Multipath mechanisms can minimize the after-challenge traffic impact on the *hot* links as well as the whole network. Furthermore, splitting traffic onto different paths strategically may potentially provide more throughput. Multipath routing has been widely studied as an effective mechanism to reduce congestion in *hot spots* by deviating traffic to unused network resources [104, 109, 110]. Several methods for load balancing using multipath routing without survivability measures have been researched. For example, optimization

has been done to maximize the flow on each path in the ECMP routing algorithm [111]. Another optimization problem has been formulated by a weighted multipath routing based on ECMP; its objective function is to minimize the maximum link utilization [112]. Resilient Overlay Networks [113, 114] has been studied to improve the robustness and availability of current Internet paths. However, from a traffic engineering perspective, multipath routing is advantageous for small networks only for the all-commodity traffic scenario, yet the multipath gain diminishes as the network becomes large [115].

When multiple next-hop addresses are installed in the routers' forwarding table, scheduling mechanisms have to be redesigned. Multi-Topology Routing (MTR) constructs multiple topologies with different link weight configurations and enables separate forwarding mechanism on a per-topology basis. It is a simple mechanism to perform and each packet can switch among topologies [63, 116]. A source routing deflection mechanism uses tags to apply path diversity for multipath routing [64]. Another approach is to forward traffic on all paths that make forwarding progress towards the destination [117] with one set of link weights. Each router makes local forwarding decisions using a shorter-hop path towards the destination and therefore, loop-free paths are guaranteed.

From the flow-level's perspective for multiple path forwarding, three major forwarding mechanisms [62], round robin, hashing, flow caching can be used. Round robin mechanism forwards traffic among several paths in a round robin manner. A weighted round robin can prioritize different class of traffic. Little overhead is introduced to the current intradomain protocol when using this mechanism. However, there is no guarantee for packets to be mapped to the same flow which can cause packet reordering, and it slows down the transport protocol. The second mechanism is hashing. Different paths are divided into several hash ranges and packets are hashed based on their header information. This mechanism ensures in-order packet delivery since packets from the same flow has similar hash value and in turn been mapped to the same outgoing path [118]. However,

fine-grained flow distribution is not possible since the flows are not distinguished among each other. For example, elephant and mice flow can cause unbalanced load among paths. Flow caching is a widely adopted approach for multipath forwarding by combining the benefits of round robin and hashing. Packets from a previous cached flow are forwarded to the same path while packets from a new flow can be forwarded on any previous path to achieve fine-grained flow distribution. An extension to flow caching is flowlet cache [119], which defines that if the time between two successive packets is larger than the maximum delay difference between the paths, it can be safely forwarded on any available path without causing reordering.

2.3.3 Resilient Multipath Architecture

Most of the traffic allocation and multipath routing studies assume normal network connectivity or single link failure [109, 120]. This is widely studied and considered as an effective resilient multipath routing mechanism for a single link failure [121, 122]. Few have considered geo-correlated challenges, in which the traffic allocation follows the widest paths disjoint with respect to the bottleneck links. The bottleneck links from multiple paths are mutually disjoint to increase resilience [123]. An optimization problem has been formulated to model the issues in a multi-source-destination routing environment, and it leads to a pseudo-polynomial algorithm based on linear programming with a bounded buffer size and skew constraint [124]. For path skew analysis in the multipath routing context, past work calculates a number of shortest paths and selects the ones that meet the skew requirement. The returned paths are then used to solve the optimization problem [124]. A multipath flow optimization problem has been formulated with two objectives, total link utilization and bandwidth fairness, and has been solved with a nonlinear programming solver [125]. However, with the increasing importance of network resilience under large-scale failures or attacks, it is imperative to analyze

multipath routing efficiency and understand the traffic allocation requirements under these challenges.

Several resilient transport protocols have been proposed. mTCP [126] can aggregate the bandwidth of several paths concurrently and improve resilience using the redundant path. Multipath transfer (CMT) can distribute data across multiple end-to-end paths in a multi-homed devices to achieve efficient parallel data transfer [127]. Multi-Path TCP (MPTCP) [25, 26] enables simultaneous use of several network interfaces to establish multiple subflows for a host pair. It provides better throughput and survivability to failures while preserves the regular TCP interface to applications. However, there is no control over how the multiple paths for different subflows are calculated. ResTP [22,128,129] is a resilient general-purpose transport layer protocol. By employing a set of reliability mechanisms that are composable and tunable, it is flexible in efficiently supporting various application classes operating across different network environments with distinct characteristics. It establishes multiple transport flows for its data transmission by taking advantage of the geodiverse path set and the traffic allocation information provided by the GeoPath Diverse Routing Protocol GeoDivRP; ResTP can either actively spread the data over all available paths to survive a single path failure with no disruption or transmit the data on one path while leaving another as a hot-standby for rapid failover. This dissertation considers only the multipath mode in which all paths transport traffic. In addition to multipath spreading capability, ResTP also provides other transport-layer services to the application layer, including multiplexing/demultiplexing, adaptive flow/subflow management, flexibly composable error control, and flow control and congestion control. As noted above, with the goal of supporting a variety of application types, each of these services is comprised of multiple composable mechanisms [130]. It chooses among its various reliability mechanisms to satisfy the specific application it is servicing according to the particular mission requirements.

2.4 Network Optimization

Network optimization is a popular research topic for modeling and designing of computer networks. Numerous research has been done for better link utilization, load balancing, and network resilience. For example, in intradomain traffic engineering, a common approach is to minimize the maximum link utilization [131]. There are other approaches, such as MPLS networks [132], and WDM networks with optical cross-connects [133]. Delay minimization is widely studied for network communication [120, 134–136]. Jitter and delay minimization optimization has been proposed [137] for multimedia applications. The M/M/1 queuing model is widely used for network delay [137, 138].

Resilient routing can benefit from the network optimization. Diverse path routing has been proposed to guarantee multiple correlated failures and to form two paths with the smallest joint failure ratio through an optimization formulation [68]. The network flow inhibition problem has been proposed with the objective to minimize the maximum flow in the graph with a given attack budget [139–142]. Most communication networks with optimization constraints use multi-commodity flow-problem formulation due to the fact that multiple node pairs are communicating at the same time.

2.4.1 Multi-Commodity Flow Problem

The multi-commodity flow (MCF) problem is to achieve a certain objective when multiple flow demands exist among several source–destination node pairs or commodities. Communicating over the shortest paths for each commodity can achieve minimum delay or transmission cost in the network communication with no bandwidth constraints. However, the constraint exists in all real-world topologies. This problem arises from the fact that there are multiple demands to be fulfilled in the network simultaneously and they compete for the limited network resources. The *multi* comes from the requirement

of multiple commodities required to communicate at the same time. We can mathematically represent an MCF problem in terms of the flow variables x_e^w defined as the amount of flow for the commodity w transmitting over the network link (e).

Assume there are W commodities defined by $W^w = (s^w, t^w, m^w)$, where s^w and t^w are the source and destination of the commodity w, and m^w is the traffic demand. We use the link-path formulation [138,143] of the problem and further present a linear and a nonlinear programming formulation to solve both of the optimization problems, respectively. We are not using the other popular link-node formulation [138, 143] since GeoDivRP needs to dictate the distance between different paths, and the Link-Node formulation would introduce extra complexity for calculating paths.

2.5 Software-Defined Networking

Software-Defined Networking (SDN) [144] is a concept of using programmable components to control network behaviors. Resilience services require constant monitoring and remediation tasks; SDN can potentially support these resilience services. By dividing the network control and data functions, network services are abstracted from the underlying infrastructure. This enables rapid innovation as new versions of network software can be easily deployed. OpenFlow [145] is the first open standard southbound interface for realizing SDN. It achieves flexible and programmable data transmission through defined actions for each flow entry; the actions include packet forwarding, packet drop, and traffic shaping. OpenFlow has a tiered architecture in which the southbound interface directly controls the network devices, and the northbound interface presents abstraction to the application for easier development. Network experiments using physical OpenFlow-enabled switches are usually difficult to carry out. Mininet [28] is used for network emulation and proof-of-concept design precede real-world deployment. A Mininet configuration frame-

work has been proposed to accelerate the experiments cycle with the ability to test with real-world topologies [146]. A systematic design using OpenFlow framework to build disaster-resilient network has been proposed [147].

2.6 Summary

From the literature review, we conclude that geo-correlated challenge poses great threat to dependable network communications. Mechanisms have been proposed to improve network resilience. However, a framework to fully understand the challenges' characteristics and a full-stack protocol to improve traffic transmission resilience are required; these two areas are this dissertation's main focus.

Chapter 3

Topology Vulnerability Analysis

In this chapter, we present a systematic analysis of the topology vulnerability to geo-correlated challenges. We define the geodiversity and propose a graph metric for evaluating the resilience of a given topology against geo-correlated challenges in Section 3.1. We design our critical-region identification mechanism and employ multiple fiber-level network topologies in different continents to verify its effectiveness in Section 3.2. In Section 3.3, we study the implication of malicious attacks using centrality metrics and provide performance analysis when a restoration mechanism is in place.

3.1 Geodiversity Definition

This section presents a formal definition of the geodiversity metric and its aggregate properties when applied to each node pair as well as to the complete network graph. It is an extension from link/node-disjoint diversity by considering geographic diversity between different paths. We evaluate geodiversity based on its ability to reflect the underlying graph's connectivity, and propose a graph metric for differentiating the geodiversity of different physical topologies.

Geographic diversity $D(p_a)$ such that $D \geq d$ is defined as the minimum distance

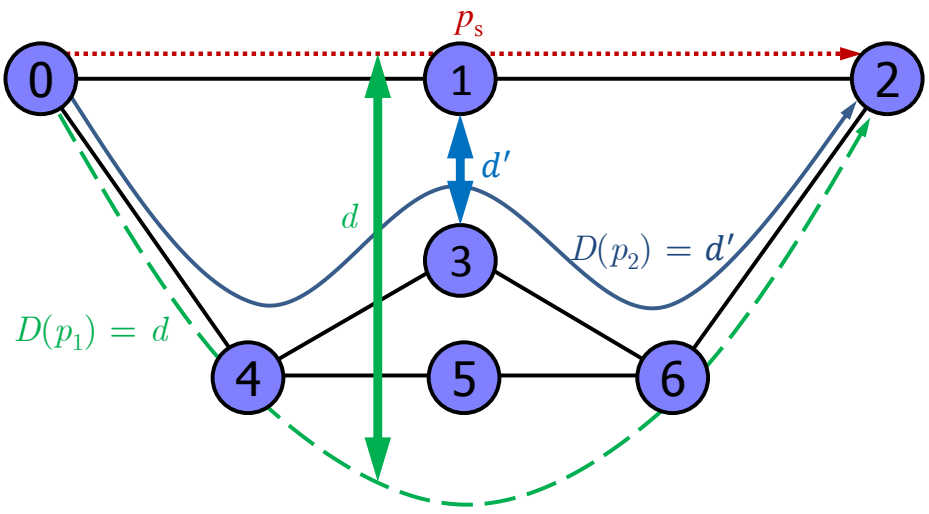

Figure 3.1: Geographic diversity: distance d

between any node members of path vector p_a and that of the shortest path. Based on the shortest path p_s with a given distance-separation criteria d, the qualified path p_a is defined as a GeoPath. The set of distance d-separated paths for a given node pair is the GeoPaths. Consider Figure 3.1 in which node 0 is the source and node 2 is the destination node and there are three paths in total. The red dotted line shows the shortest path p_s consists of nodes 0–1–2. The green dashed line shows path p_1 and its geodiversity $D(p_1)$ (with respect to p_s) equals d. The blue solid line shows path p_2 and its geodiversity $D(p_2)$ is d' since the minimum distance is d' between node 1 and node 3.

Based on the geographic diversity, we start the Effective Geographical Path Diversity (EGPD) metric calculation by taking weighted additional diversity from added GeoPaths based on previous path diversity work that didn't consider geographic diversity [22]:

$$\text{EGPD} = 1 - e^{-\lambda k_{v_s v_d}} \tag{3.1}$$

where λ is an experimentally determined constant that scales the impact of $k_{v_s v_d}$ based on the utility of this added diversity, while v_s is the source node and v_d the destination

node. $k_{v_s v_d}$ is the sum of all non-zero diversity paths defined as:

$$k_{v_s v_d} = \sum_{i=1}^{m} D(p_i), \tag{3.2}$$

The range of EGPD is between $[0, 1]$ where 0 means that there is no diversity in the graph as there is no alternative path connecting any pair of nodes. When EGPD approaches 1, geographic diversity increases; the value 1 means that for any node pair in the graph, there exist at least two GeoPaths based on the given distance-separation criteria d.

The Total Graph Geographic Diversity (TGGD) is simply the average of the EPGD value for all node pairs within that graph similar to the past work [21, 22]. Therefore, this metric is an important factor for resilience of the network topology in face of geo-correlated challenges. Based on the TGGD calculated, we obtain the compensated TGGD value as follows[1]:

$$\text{cTGGD} = e^{\text{TGGD}-1} \times (\frac{\|G_M\|}{\|G\|})^{-\rho}. \tag{3.3}$$

Here, $\|G\|$ is the total number of links in topology G, and $\|G_M\|$ is the total number of links for the largest network topology in consideration (in this case 244 links for AT&T). We weight the graph diversity based on the ratio of $\|G\|$ and $\|G_M\|$. The purpose of the weight is two fold. First is to eliminate the penalty to a dense network as it has less geographic diversity for any node pair within a given region; this is because the links are not able to be as separated geographically as a sparse network. Second, it is normalized by the number of links of the largest topology in the comparison set; therefore, cTGGD indicates the relative resilience level of topologies against the largest topology. The tuning parameter ρ is experimentally chosen as 0.05.

cTGGD represents the resilience level of a certain topology through the incrementally added GeoPath. We present the metric comparison results in the next section and show

[1]Based on past work [21, 22] with modification to incorporate geodiversity

cTGGD to be a good indicator for the network resilience in face of geo-correlated challenges. With carefully selected ρ value, cTGGD efficiently distinguish the geodiversity among topologies.

The complexity of cTGGD depends on the algorithm used to calculate GeoPaths. If the iterative WayPoint Shortest Path (iWPSP) heuristic introduced in Chapter 4 is used, each path for a node pair has a best-case complexity of $O(|V| \log |V|)$. With most of the physical topologies having an average degree below a constant number four [24], the best-case complexity for cTGGD is $O(|V| \log |V|)$ and the worst case is $O(|V|^2 \log |V|)$.

3.1.1 Flow Robustness

Before explaining the identification mechanism in detail, we introduce the network performance metric used in this chapter, *flow robustness* [22,148]. A flow is considered *reliable* if at least one path remains connected during the failure. We compute flow robustness to be the number of reliable flows divided by the number of total flows that exist in the network [21]. Link and node removal based on a fixed probability of failure have been analyzed [22]. We consider regional challenges, where a challenge fails nodes covered by the region, along with links connected to the failed nodes. The algorithmic complexity depends on the time to find the number of components in a given graph, which makes the complexity as $O(|V| + |E|)$. We use flow robustness in this work for two reasons; first it matches the packet delivery ratio (PDR) result in network simulations for all node pairs communicating with constant bit rate (CBR) traffic, for example, the PDR result in Chapter 4 using CBR traffic is related to flow robustness in the simulation context. Second it is effective and efficient in terms of evaluating the network connectivity.

A related metric, all-terminal reliability [149], calculates the probability that a given node pair can communicate with each other for a given period of time. However, flow

robustness considers the connectivity of a given node pair at any instance of time; it is efficient in the scenario of this work since we are concerned with instantaneous connectivity. Furthermore, all-terminal reliability requires a connected graph.

3.2 Critical Region Identification

Given the definition of geodiversity, the ability to pinpoint critical regions against geo-correlated challenges becomes important for an efficient network restoration mechanism. We propose a *critical-region identification* mechanism to identify vulnerable areas in a topology and suggest some counter-measures to prepare the current Internet for either failures or attacks. We further incorporate the identified critical regions into a mechanism to analyze the relative resilience among different topologies and compare it with cTGGD.

3.2.1 Identification Mechanism

We propose a critical-region identification mechanism using the *minimum-covering circle* algorithm [150] and verify its effectiveness in recognizing vulnerable areas. We find the critical region by assuming a circular failure with a given epicenter and a radius. This is one endeavor to help analyze the design and maintenance of the normal network communication in face of challenges.

With the network defined as $G = (V, E)$. A circular region f is defined as the circular area with the failure center c and the radius r:

$$f = (c, r) \tag{3.4}$$

We further define s_{c,v_i} as the distance from node v_i to the failure center c. The *challenge*

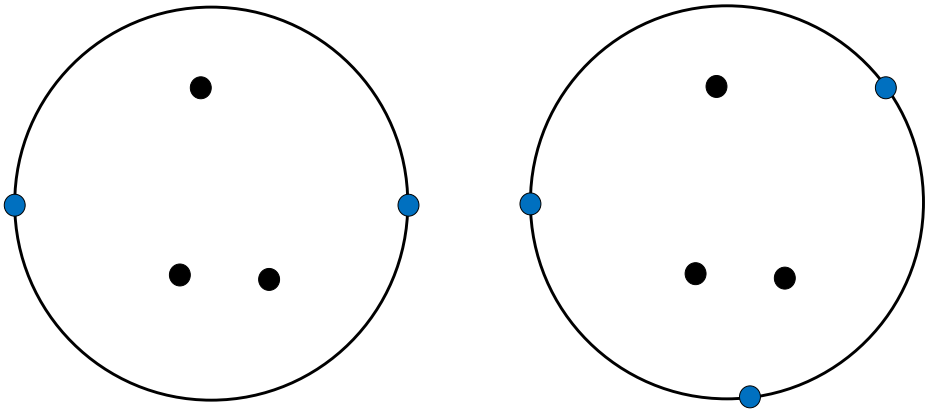

Figure 3.2: Smallest-enclosing circle problem

node set of a given failure f is the node set V that qualifies the following condition:

$$V|s_{c,v_i} \leq r \tag{3.5}$$

In other words, the challenged node set is the set of nodes that can be covered by the failure f. Any node within the circle will be disrupted and removed from the connectivity calculation, along with its connected links, of course.

The objective of the identification model is to find the smallest circle that covers the challenged node set; with whose removal can the flow robustness drop below a target. The evaluating graph metric can be any other global measures such as the network efficiency or the giant component size.

The smallest-circle problem originates in the fact that the minimum-covering circle of a node set can be determined by at most three points and they have to lie on the rim of the circle [150]. The points considered in the smallest-circle problem can be considered as network nodes and the covering circle the challenge area. As shown in Figure 3.2, the minimum circle to cover a given node set is either determined by two nodes that form the diameter of the circle, or three nodes on the rim of the circle.

By considering all the circles enclosed by two nodes and three nodes, the model obtains a list of candidate failure regions with their corresponding challenged node set covered (challenged) by the circle (failure region). By calculating the flow robustness of the topology after removing each challenged node set sequentially, we can find the minimum enclosing circle which can drop the flow robustness below a given flow robustness target.

The algorithm's complexity can be divided into two parts. The first one is the identification of every possible failed region, which is $O(|V|^3)$ with $|V|$ representing the number of nodes. It is the complexity of finding candidate failure circles using three nodes. The second part that calculates the flow robustness after each circular challenge is $O(|V|+|E|)$ with $|E|$ representing the number of links. Since $|E|$ is generally in the same magnitude of $|V|$ for fiber-level networks, the complexity is reduced to $O(|V|)$. Therefore, the overall complexity of the identification model is $O(|V|^4)$. Since the identification is deterministic for the fiber-level networks, given the slow deployment of new fibers, it can be easily calculated for most of the considered topologies with the number of nodes $|V|$ in the scale of hundreds [42].

By averaging the list of flow robustness results obtained from the identification mechanism, we can evaluate the relative resilience of different network topologies. We define aggregated remaining flow (ARF) for a given topology and its value is in the range $[0, 1)$.

Aggregated Remaining Flow is defined as the average flow robustness after each *challenged node set* is removed

$$\text{ARF} = \frac{\sum_{i=1}^{n}(\text{FR}_i)}{|V|} \tag{3.6}$$

where FR_i is the flow robustness target for a given challenge i, $|V|$ is the number of challenged node set identified in a given topology. The larger the ARF is, the more

robust a certain topology is against area-based challenges. We further propose normalized aggregated remaining flow (nARF) as follows.

Normalized Aggregated Remaining Flow is defined as the remaining flow robustness after a list of nodes are failed, normalized by the total number of links.

$$\text{nARF} = e^{\text{ARF}-1} \times (\frac{\|G_M\|}{\|G\|})^{-\rho} \tag{3.7}$$

$\|G\|$ is the total number of links in topology G, and $\|G_M\|$ is the total number of links for the largest network topology in consideration (in this case 244 links for AT&T). The minimum-covering circle model produces a limited number of challenged node sets. Since both ARF and nARF are calculated after removing each challenged node set sequentially, they are time-bounded.

3.2.2 Numerical Results

We analyze the fiber-level topologies from different continents. We include Level 3 [151] and Sprint [152] networks for the US, and the Bestel network [153] for Mexico. For European topologies, we include Oteglobe [154], LambdaNet [155], and NORDUnet [156]. Oteglobe is based in Europe and serves as one intracontinental network.

For the critical distance comparison, we further include US topologies such as AT&T [157], Internet2 [158], TeliaSonera [159], and CORONET [160] networks. CORONET is a synthetic fiber network to represent Internet service provider topology. SUNET (Swedish University Computer Network) [161] is included as an European research topology.

North American Topologies

We start by presenting the critical region result for the Level 3 network in Figure 3.3. The flow robustness target is shown in different color shades to represent the varying

vulnerable levels. The circles shown are the minimum failure regions to reduce the flow robustness of a given topology below the given target. The darker color shade represents a larger flow robustness target, and the better the network performs. All the critical regions are in the northeast corner of the topology. The critical regions for the larger flow robustness target concentrate around New York City and gradually shift in the southwest direction as the it becomes smaller. For example, when flow robustness target is 0.9, the critical region centers at New York, NY, and shifts to Butler, PA as it becomes 0.6. This is because for the larger flow robustness target, the most effective location is around New York City as it has a more dense network component concentration; and for the smaller flow robustness target, the failure regions center around Pennsylvania and can efficiently disrupt the connection between the east and the west coast as it is a narrow corridor for the US topology.

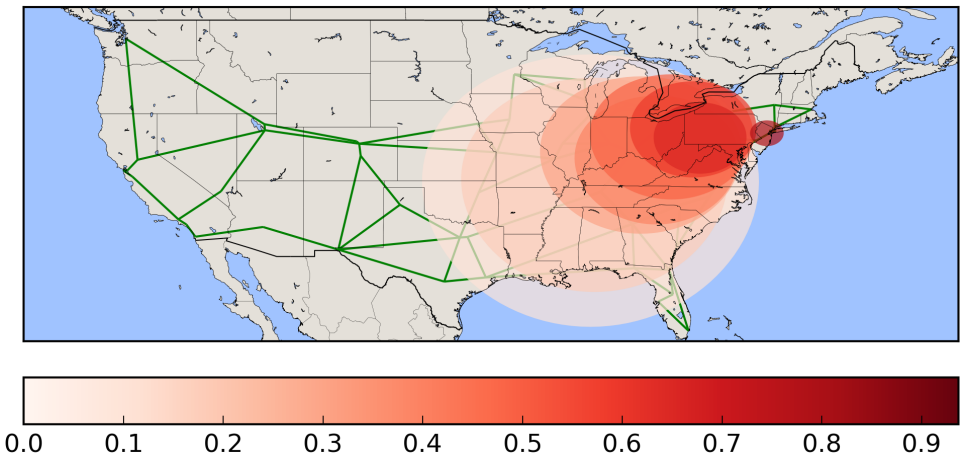

Figure 3.3: Level 3 unweighted network critical region analysis

However, when we introduce population-based weighted topologies, the critical region shifts to more populated areas. As shown in Figure 3.4, with a larger flow robustness target, the critical region shifts from the northeast corner of the topology for the unweighted graph to around Chicago. For example, the failure region for the unweighted graph centers at Butler, PA when flow robustness target is 0.6, yet it moves toward Van

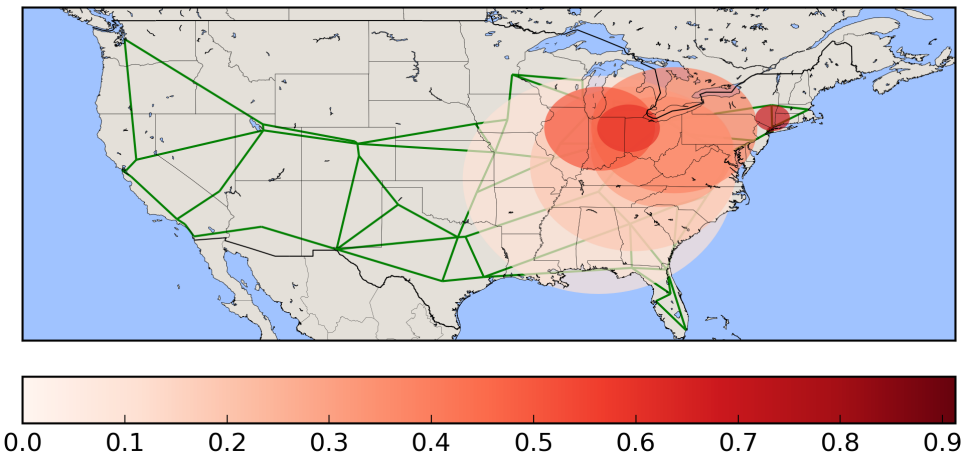

Figure 3.4: Level 3 weighted network critical region analysis

Wert, OH for the weighted one. This is because the Chicago node contributes more weight to its adjacent links due to its large population.

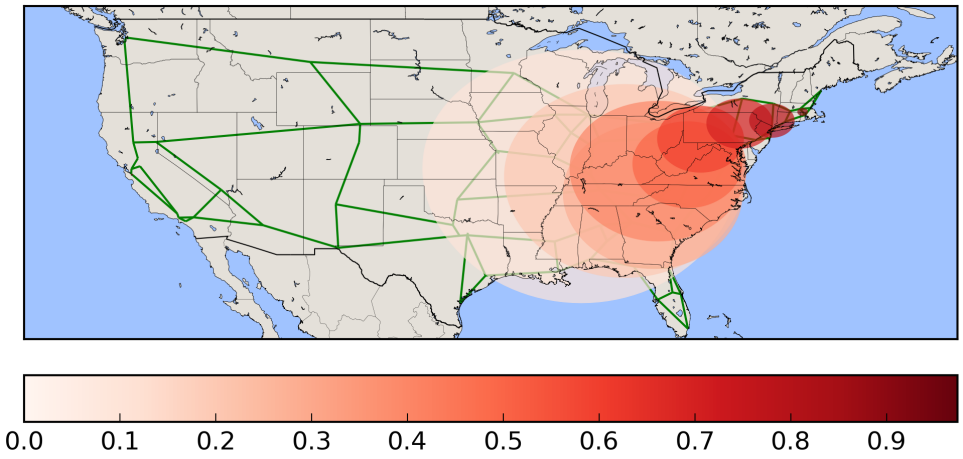

Figure 3.5: Sprint unweighted network critical region analysis

We further present results for the unweighted Sprint network in Figure 3.5. It presents a similar result to Level 3. To achieve the same flow robustness target, the Sprint network has a comparatively smaller circular region due to its more concentrated network nodes and links than Level 3.

For the weighted graph as shown in Figure 3.6, the degree of shifting towards Chicago

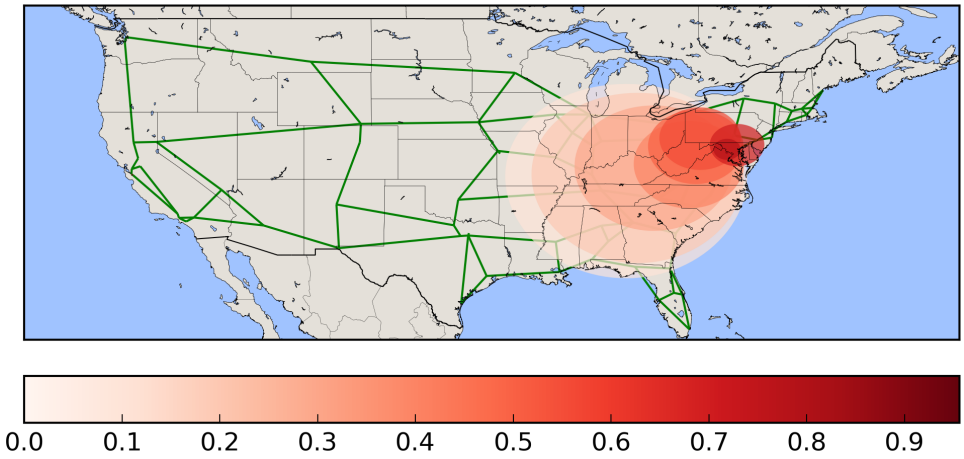

Figure 3.6: Sprint weighted network critical region analysis

node for Sprint is smaller than Level 3. This is because the Sprint network has some highly-populated nodes around West Virginia, Virginia, and Kentucky which Level 3 lack.

We carry out similar analysis for the the Bestel [153] network, one of the largest telecommunication networks in Mexico. As shown in Figure 3.7, the critical regions for different flow robustness target values are spread out. For the larger flow robustness target, the failure radius is fairly small and it affects only a single node on the edge of the topology. As the flow robustness target decreases, the failure region grows larger and most of the critical regions center around Mexico City.

European Topologies

Similar analysis is carried out on European topologies. We begin with Oteglobe [154], an international carrier which is strong in the southeast Europe. As shown in Figure 3.8, the critical regions focus around Greece as it is the network headquarter with a higher degree. The network is more resilient to regional failures since the network spans across a wider geographic region and the topology is relatively sparse compared to the US carriers.

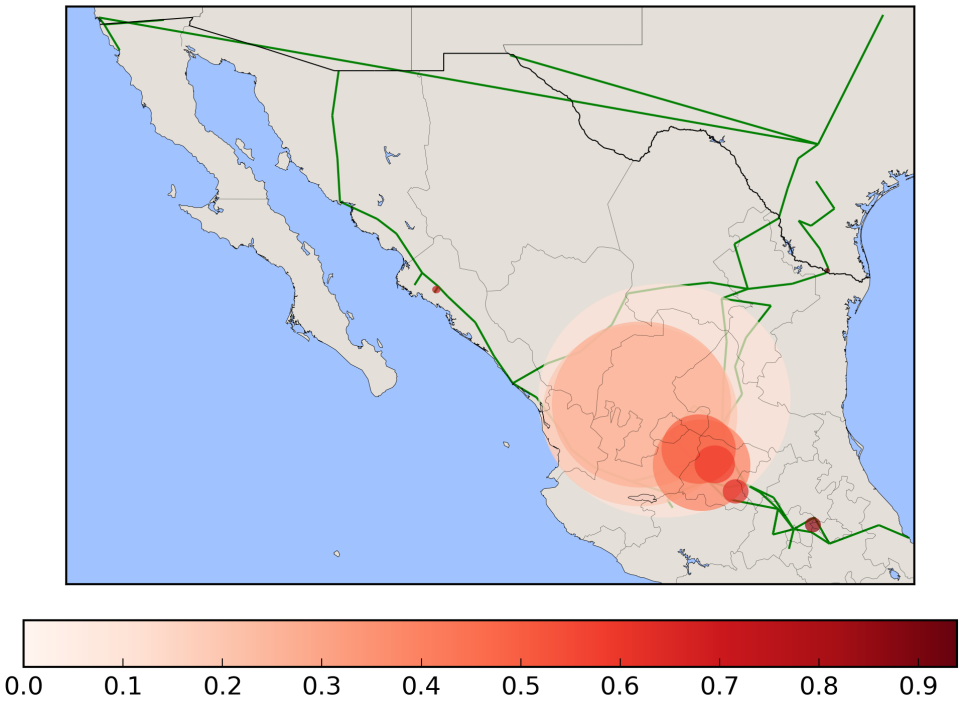

Figure 3.7: Bestel network critical region analysis

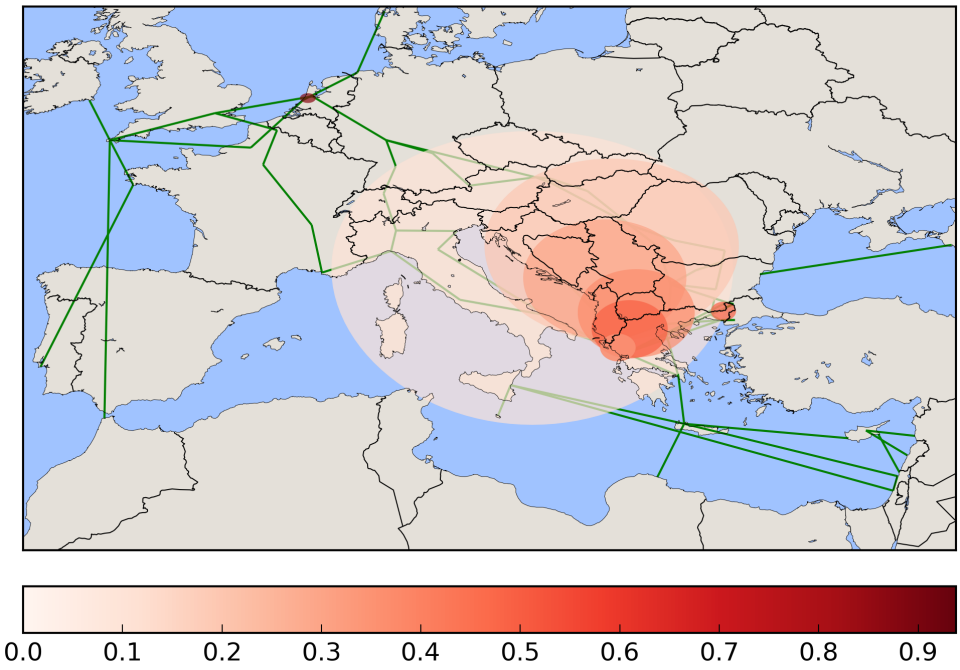

Figure 3.8: Oteglobe network critical region analysis

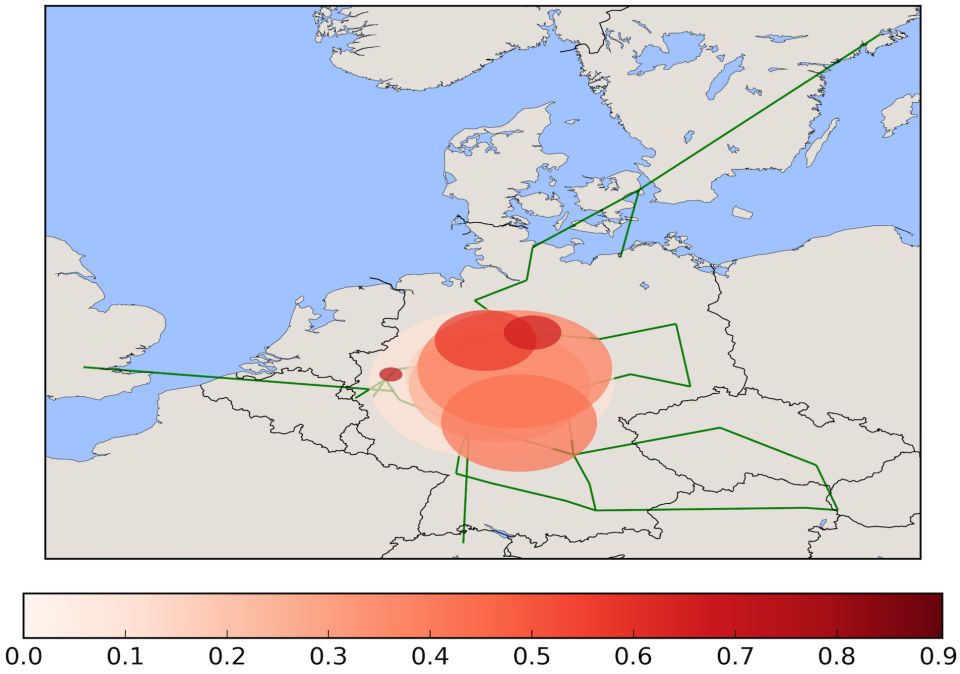

Figure 3.9: LambdaNet network critical region analysis

LambdaNet is a network topology owned by euNetworks [155] and it lies mostly in Germany. Contrary to other large-scale networks, it is a regional and relatively smaller network. As shown in Figure 3.9, the failure regions focus around the geographical center of Germany.

Critical Distance Comparison

We present the critical distance results for the US networks in Figure 3.10. As the flow robustness target increases, the failure radius decreases. Surprisingly, we observe that all the topologies have similar critical distances; this means that to achieve a similar damage to the considered US networks, a similar scale of challenge is required. It implies that all the US networks present similar properties in the face of geo-correlated challenges, even though the networks have different characteristics, such as different number of nodes and links. This is mainly because that most fibers are deployed along the US motorways [13],

and have a similar resilience against a similar level of geo-correlated challenge. This also means that for a given threat model, the protocol design is the same for all the US topologies. We introduce the detailed protocol design in Chapter 4 and 5.

The weighted graphs result is shown in Figure 3.11. Contrary to the unweighted ones, the weighted graphs require smaller failure radii to reduce the network connectivity to a similar level. This is because the most populated nodes are located in the east coast and the critical regions are mostly there already for the unweighted topologies. The same failure region causes more damage since the nodes and links have greater weights. Similar to the previous figure, the weighted graphs present similar failure radius for the same flow robustness target. This further verifies our previous claim that different physical graphs in US have a much larger similarity than we have expected.

Overall, we list the critical failure distances for different continents in Figure 3.12. To reduce the flow robustness to 0.1, the failure radius is 600 km for Oteglobe while 170 km for LambdaNet. This is because the Oteglobe network spans across multiple countries and covers a wider geographical area.

We further include the detailed vulnerable locations for the flow robustness target of 0.6 in Table 3.1. The locations are centered around Virginia and Pennsylvania; this is because if the challenges occurred in these locations, most of the northeast US will be disconnected from the rest of the network. Note that the center of the failure is not necessarily at a particular node in the topology.

Furthermore, we present our suggestion for prioritized protection nodes to increase the overall resilience level against geo-correlated challenges for various US topologies. As shown in Table 3.2, for different flow robustness (FR) targets, the critical region has a number of enclosed nodes (cities), and we choose the node (city) with the highest degree centrality as the prioritized protection node for that region; the reason for using

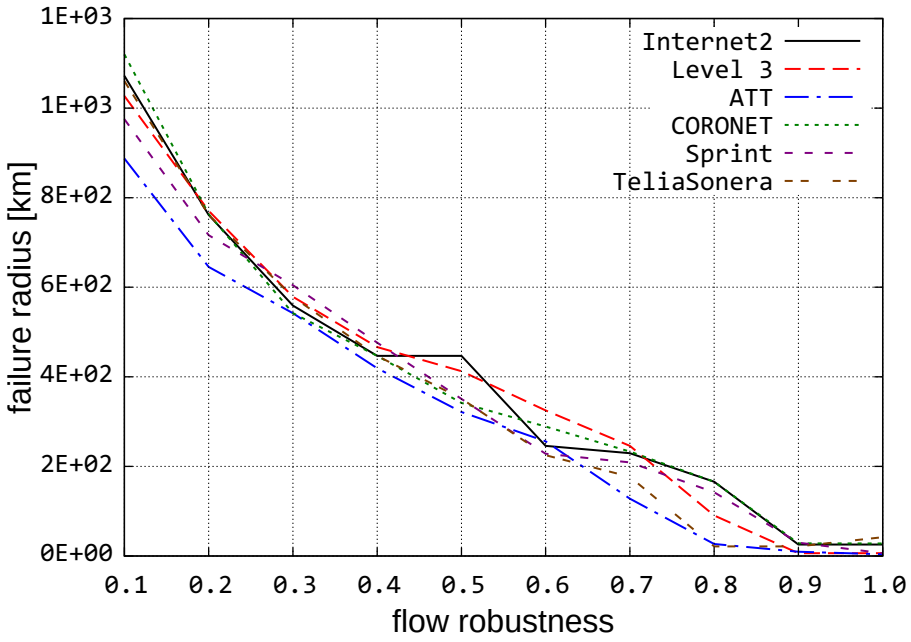

Figure 3.10: Challenge distances for unweighted US graph

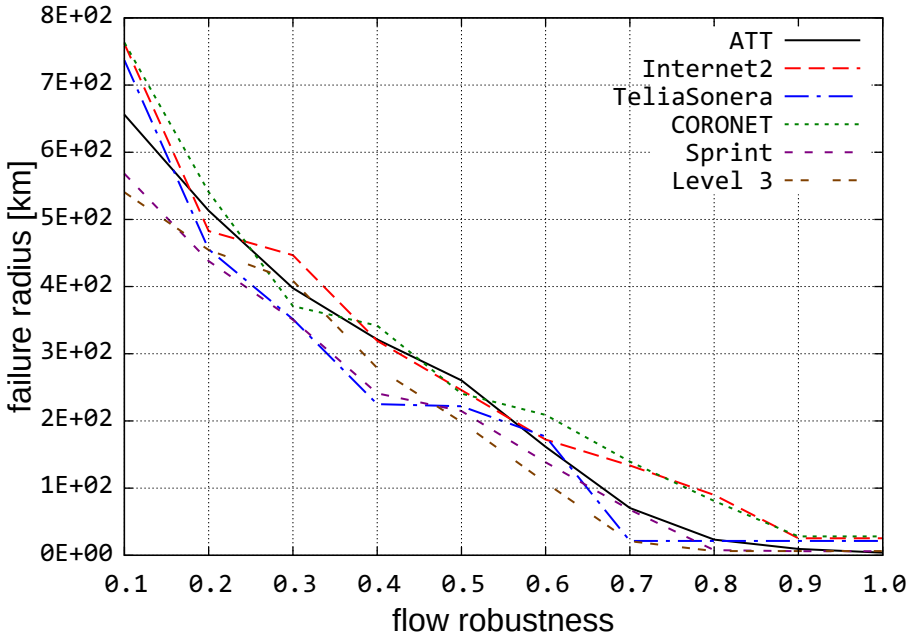

Figure 3.11: Challenge distances for weighted US graph

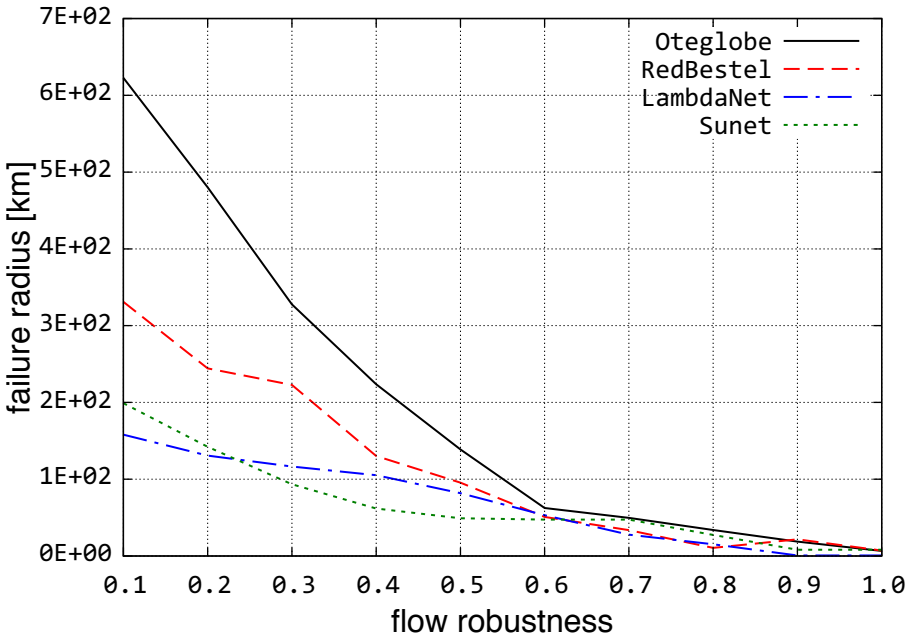

Figure 3.12: Challenge distances for different continents

the degree centrality [45, 46] as the selection measure is that the high degree centrality node (city) serves an important role during geo-correlated challenges, which we present in the following section.

We apply this mechanism to verify and present a comparison of our graph metrics in Table 3.3. As shown in this table, the difference among topologies in terms of both TGGD and ARF is minimal. This is because all the topologies are designed with nodes and links separated; which provides survivability against area-based challenges. However, the topologies have different number of links and his analysis has a penalty against dense networks. Both the cTGGD and nARF metrics eliminate this effect after normalization based on the number of links. cTGGD and nARF have shown comparable results by successfully distinguishing geographic diversity levels in topologies. The numbers in bold are the two topologies with the largest cTGGD and nARF values. nARF has a higher computational complexity since the critical-region identification mechanism has

a complexity of $O(|V|^4)$. The cTGGD metric effectively indicates the resilience level while at the same time having a substantially lower complexity compared to nARF. The best-case complexity for cTGGD is $O(|V| \log |V|)$, while the worst-case complexity is still only $O(|V|^2 \log |V|)$.

3.3 Malicious Attacks

In addition to understanding the challenges from large-scale disasters, we further explore how targeted malicious attacks can affect physical layer networks in this section. An example of such attacks could be an electromagnetic pulse (EMP) weapon. From the standpoint of the attackers, we analyze the mechanisms they could exploit to increase the damage using a given attack budget. We assume the cost to increase the attack area is proportional to the budget, which means that the radius of the attack corresponds to the square root of the budget. We use cost b to represent the cost to fail an area with a radius r in the physical topologies, while the number of attack locations a corresponds to the number of challenges that share the total attack budget.

Cost Radius Relation is defined as the radius of each attack location, given the attack budget (b) and the number of attack locations (a)

$$r = \sqrt{\frac{b}{\pi a}} \tag{3.8}$$

We apply different centrality-based attacks on several network regions to study which metric generates the worst damage to a given topology. We employ several best-known centrality metrics: betweenness, closeness, eigenvector, load, and degree centrality [45,46] to analyze different physical networks. Our model provides a list of nodes sorted according to their different centrality values from high to low. The definition of these centrality

metrics are provide in Chapter 2. These centrality metrics have been used to study the performance of networks against targeted malicious attacks [19, 41].

The attack starts with a challenge area defined by the *cost radius relation* (CRR) centered at the highest centrality node identified from the previous step. Given a fixed budget b and the attack can occur in one location or divided into multiple locations each with a smaller radius r. For simplicity of the analysis, we assume that the malicious attacks in different locations are divided equally in terms of area. For example, if the total challenge area is ten and the number of challenge locations is two, then each challenge location has an area of five. We present how the number of attack locations affects the overall flow robustness for the AT&T physical network in Figure 3.13. As the number of challenge location increases, the flow robustness value decreases. For example, the degree centrality attack drops the flow robustness to below 40% when the number of locations is 16. Furthermore, after the challenge locations increase beyond four, the value of flow robustness stabilizes. As it would be more complicated and susceptible to detection for the attackers to increase the number of attack locations, we conclude that by dividing the attacks into four locations and deploying them based on the higher degree centrality maximizes the attack damage in the AT&T network.

Figure 3.14 shows similar results when the attacks occur in the Sprint network. Similar to the AT&T network, degree centrality has the greatest impact to the flow robustness. However, the significant drop in flow robustness occurs around eight challenge locations. This is partly due to the evenly distributed network nodes and links in the Sprint network. However, when the number of locations increases beyond eight, the flow robustness drops significantly, which is because the network is partitioned after the higher centrality nodes have been removed.

Figure 3.15 presents the Level 3 physical network under attacks. It shows faster and

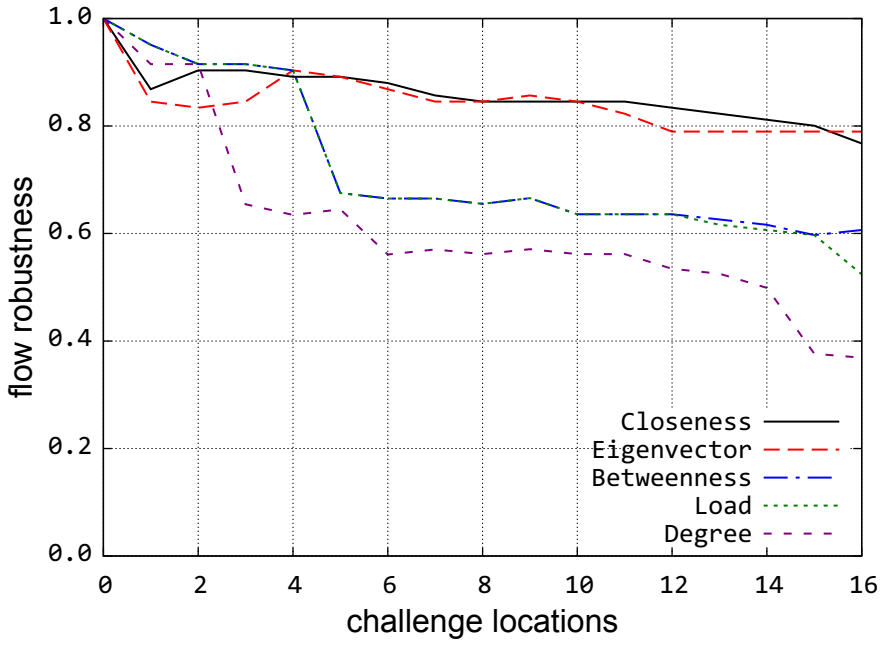

Figure 3.13: AT&T optical network under regional challenges

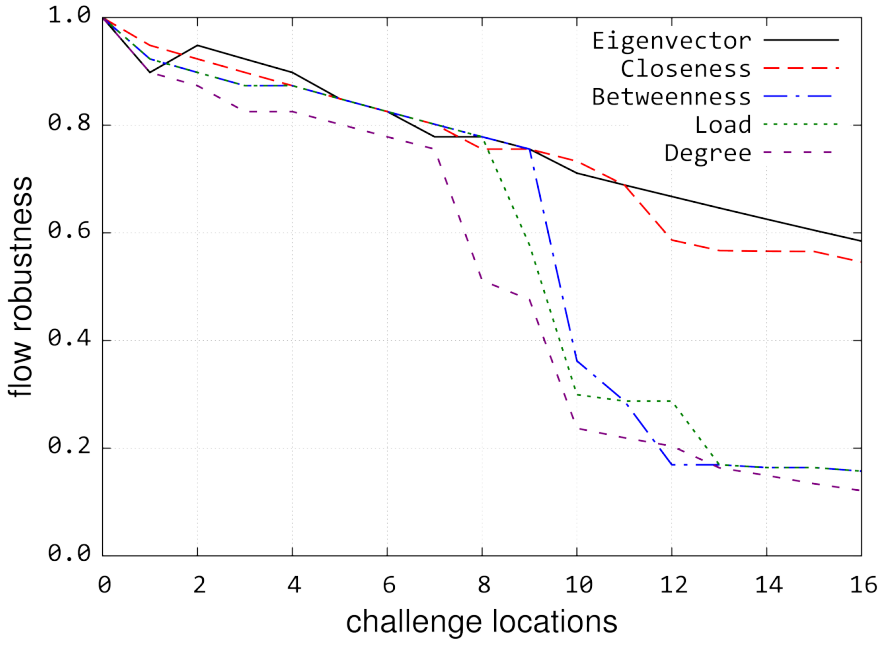

Figure 3.14: Sprint network under regional challenges

more significant damage than the previous cases. When the number of challenge location increases to eight, the flow robustness drops below 20%. This attack result demonstrates that with a certain amount of knowledge of the network topology and expertise to analyze it, attackers can cause a substantial amount of damage even with a small budget.

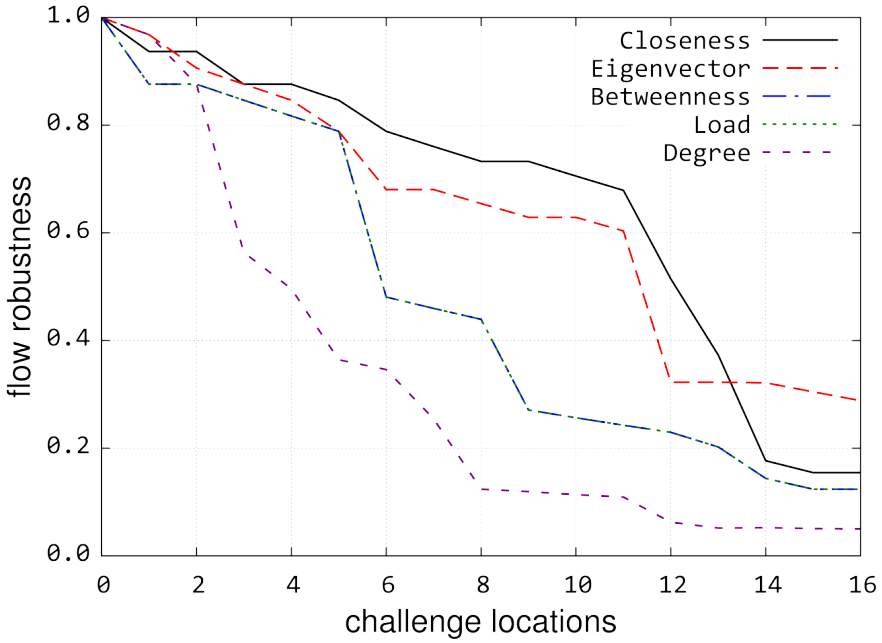

Figure 3.15: Level 3 network under regional challenges

Similar analysis is performed in several other topologies and shown in Appendix A.

3.3.1 Restoration Mechanism

Network restoration time can vary from a few hundred seconds to days [162]. In this section, we analyze the effectiveness of network restoration schemes and provide network improvement suggestions. We present flow robustness results when the network has a restoration plan and demonstrate its improvement when the plan is in place. This is one endeavor to better understand the challenge characteristics and suggest network design guidelines.

The result from the critical-region identification mechanism in the previous section reveals vulnerable network locations and can guide the improvement of the overall network resilience. For example, adding physical protection for existing components in the vulnerable locations can mitigate the impact of attacks or failures. Compared to analyzing the overall resilience and global optimization of networks, this is the local optimization based on the vulnerability level of each individual region.

We present the flow robustness improvement when a certain percentage of the failed nodes have remained connected due to a particular restoration or protection plan.[2] The challenge locations come from the critical region we have identified in Table 3.1. Due to the size of different networks, the number of challenged nodes in each location varies. As we notice from Figure 3.16, by protecting three nodes, all the physical networks increase to above 60% flow robustness. Protection can be done by shielding existing nodes or providing hot-standby nodes. The 20% flow robustness improvement is valuable for dependable network communications.

[2]Note that a specific restoration plan is not studied in this work.

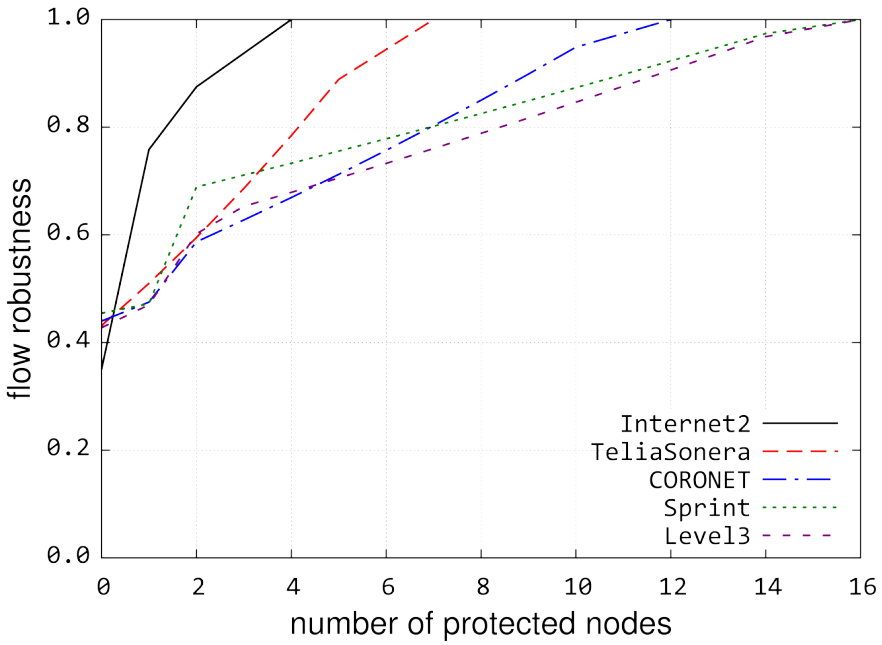

Figure 3.16: Protection plan improvement on different networks

Table 3.1: Physical topology vulnerable locations (FR=0.6)

Network	Number of Nodes	Number of Links	Flow Robustness	Challenge Centers	Challenge Coordinates	Challenge Radius (Km)	Number of Failed Nodes
AT&T	162	244	0.59	Morgantown, WV	39.67, -79.81	256	8
CORONET	39	63	0.59	West Decatur, PA	40.95, -78.32	289	6
Internet2	16	24	0.56	Stahlstown, PA	40.19, -79.35	246	2
Level 3	63	94	0.59	Butler, PA	40.84, -79.86	325	6
Sprint	77	114	0.59	Rockwood, PA	39.99, -79.27	228	6
TeliaSonera	18	21	0.53	Greensburg, PA	40.26, -79.58	225	2

Table 3.2: Prioritized protection node list

FR	AT&T	CORONET	Internet2	Level 3	Sprint	TeliaSonera
0.9	Stamford, CT	Washington DC	Boston, MA	Weehawken, NJ	Worcester, MA	Ashburn, VA
0.8	New York, NY	Washington DC	New York, NY	White Plains, NY	New York, NY	Washington DC
0.7	Trenton, NJ	Washington DC	Washington DC	Washington DC	Syracuse, NY	Washington DC
0.6	Washington DC	Washington DC	Washington DC	Washington DC	Akron, OH	Cleveland, OH
0.5	Richmond, VA	Washington DC	Nashville, TN	Washington DC	Akron, OH	Cleveland, OH
0.4	Washington DC	Raleigh, NC	Nashville, TN	Toledo, OH	Akron, OH	Cleveland, OH
0.3	Toledo, OH	St. Louis, MO	Nashville, TN	Toledo, OH	Akron, OH	Washington DC
0.2	Toledo, OH	St. Louis, MO	Nashville, TN	Toledo, OH	Chicago, IL	Kansas City, MO
0.1	Waycross, GA	Washington DC	Nashville, TN	Washington DC	Dallas, TX	Washington DC

Table 3.3: Network characteristics

Network	**TGD**	**cTGD**	**TGGD**	**cTGGD**	**ARF**	**nARF**
AT&T	0.90	0.06	0.99	**0.96**	0.86	**0.87**
CORONET	0.93	0.16	0.99	0.89	0.90	0.84
Internet2	0.88	0.26	0.94	0.81	0.88	0.80
Level 3	0.89	0.10	0.97	0.90	0.87	0.82
Sprint	0.91	0.08	0.98	**0.92**	0.87	**0.86**
TeliaSonera	0.75	0.15	0.87	0.75	0.87	0.75

Page left intentionally blank.

Chapter 4

Path Geodiverse Problem

In this chapter, we consider the *path geodiverse problem* (PGD) and provide two efficient heuristics to provide solutions. In Section 4.1, we introduce GeoPath Diverse Routing Protocol (GeoDivRP) to solve the PGD problem and present its implementation detail. We further present the simulation results using UDP traffic in Section 4.2.

4.1 GeoDivRP Implementation

We begin our discussion by introducing an optimal algorithm for the path geodiverse problem (PGD) and propose two heuristics for solving it more efficiently. PGD is defined as finding a number of GeoPaths in a given network topology. We further incorporate the heuristics in GeoDivRP and provides extensive simulation results to verify its performance.

4.1.1 GeoResLSR–Optimal Algorithm for PGD

We first formulate the path geodiverse problem (PGD) for calculating k GeoPaths and provide a two-step optimal algorithm for solving it. GeoPath is defined as the qualified distance d-separated path. Specifically, we consider PGD that involves obtaining a

set of distance d-separated paths from each and every node pair in the network. This algorithm begins with the Suurballe's algorithm [59, 60] in which the shortest-path algorithm (SPA) is iteratively applied. After each iteration of the SPA, the link weights from the constructed path is penalized by adding a factor. Once the algorithm has identified n paths, it selects the path with distance d-separated by iteratively comparing the distance between each and every node pair from all the candidate paths. Based on this algorithm, we have designed the Geodiverse Resilient Link-State Routing (GeoResLSR) protocol. This mechanism guarantees in choosing the best d-separated paths assuming a large number of candidate paths is provided. However, as SPA is applied n times for generating the candidate paths before selecting the qualified ones, its time complexity is $\mathrm{O}(n|V|(|E| + |V| \log |V|))$ [163] and the computation is slow with a large n. To reduce the complexity of this algorithm, we propose two heuristics for efficiently calculating the GeoPaths. The proposed heuristics return a set of (v_s, v_d) paths from the graph $G = (V, E)$, where V is the node (vertex) set, E is the link (edge) set. Dijkstra(G, n) is the standard Dijkstra's algorithm we use to provide the shortest path. We list the graph notations used in this section in Table 4.1.

Table 4.1: Notations for GeoDivRP

	Description
$G(V, E)$	input graph $\|\|G\|\|$ with a set of nodes V and links E
v_s	source node
v_d	destination node
v_{s_k}	neighbor node chosen by source node
v_{d_k}	neighbor node chosen by destination node
k	number of requested geodiverse paths
d	distance separation between each and every node in different disjoint paths
δ	delta distance safety margin when selecting waypoint node
buffer	distance buffer to increase link weight

4.1.2 GeoDivRP – Two Routing Heuristics

In consideration of decreasing the complexity of the GeoPaths calculation, we propose two heuristics: iterative WayPoint Shortest Path (iWPSP) and Modified Link Weight (MLW) [23, 116] [1]. As shown in Figure 4.1, for the case when $k = 3$, iWPSP first selects neighbor nodes v_{s_1} and v_{d_1} that are d distance separated from source node v_s and destination node v_d, respectively. For simplicity, we assume that such nodes exist in this work; otherwise, the nodes with the greatest distance will be chosen, iterating until nodes d apart are located. Assuming the straight segment connecting v_s and v_d is S, iWPSP selects waypoint nodes m' and m'' in the opposite direction that are distance $d + \delta$ apart from the middle node m in the shortest path, where the segment $m'mm''$ interleaves with the shortest path. Dijkstra's algorithm is performed for the two branches $v_{s_1}m'$ and $v_{d_1}m'$. By connecting the shortest path returned from the two branches, the heuristic obtains the first GeoPath p_1. The same mechanism repeats for waypoint node m'' for the second GeoPath. The variable d is a user-chosen parameter based on a threat model, and δ is experimentally chosen for different network topologies to increase the probability of successfully returning a qualified path. Furthermore, δ prevents the links from different paths from interleaving and creating routing loops. By adjusting the value of δ, this heuristic selects a nearby waypoint node if the previous one fails running the Dijkstra's algorithm. The pseudo code for iWPSP is shown in Algorithm 1.

We use a 5×5 grid network to demonstrate the GeoPaths calculated by iWPSP. As shown in Figure 4.2, we highlight the calculation of two GeoPaths from the source node 21 to destination node 3. The d value is set as less than the length of the link in the grid. Once the size of the failure region increases, a corresponding larger d value is provided to iWPSP, and the calculated paths are further apart geographically as shown in Figure 4.3.

[1] MLW is a joint work lead by Deep Medhi from UMKC

Functions:
Calculate k paths from v_s to v_d separated by distance d
begin
segment S connecting v_s and v_d, with its middle point m;
choose neighbor node v_{s_k}, v_{d_k} that is at least d distance from $v_{\mathrm{s}_{k-1}}$, $v_{\mathrm{d}_{k-1}}$, respectively;
if *k is odd number* **then**
choose two nodes m_1 and m_2 that are separated by $d+\delta$ on each direction of S, where $m_1 m m_2$ is perpendicular bisector of S;
$p_1 = \mathrm{SourceTree}_{v_{\mathrm{d}} v_{\mathrm{s}}} \leftarrow \mathrm{Dijkstra}(v_{\mathrm{d}}, v_{\mathrm{s}})$;
$k-=3$;
else
choose two nodes m_1 and m_2 that are separated by $d/2+\delta$ on each direction of L, where $m_1 m m_2$ is perpendicular bisector of S;
$k-=2$;
end
$p_{m_1 v_{\mathrm{s}_1}} = \mathrm{SourceTree}_{v_{\mathrm{s}_1} m_1} \leftarrow \mathrm{Dijkstra}(m_1, v_{\mathrm{s}_1})$;
$p_{m_2 v_{\mathrm{s}_2}} = \mathrm{SourceTree}_{v_{\mathrm{s}_2} m_2} \leftarrow \mathrm{Dijkstra}(m_2, v_{\mathrm{s}_2})$;
$p_{m_1 v_{\mathrm{d}_1}} = \mathrm{SourceTree}_{v_{\mathrm{d}_1} m_1} \leftarrow \mathrm{Dijkstra}(m_1, v_{\mathrm{d}_1})$;
$p_{m_2 v_{\mathrm{d}_2}} = \mathrm{SourceTree}_{v_{\mathrm{d}_2} m_2} \leftarrow \mathrm{Dijkstra}(m_2, v_{\mathrm{d}_2})$;
while $k > 0$ **do**
segment S = newest established path;
choose one node m_k that is separated by distance $d+\delta$ from S on the farther direction from the absolute shortest path;
$p_{m_k v_{\mathrm{s}_k}} = \mathrm{SourceTree}_{m_k v_{\mathrm{s}_k}} \leftarrow \mathrm{Dijkstra}(m_k, v_{\mathrm{s}_k})$;
$p_{m_k v_{\mathrm{d}_k}} = \mathrm{SourceTree}_{m_k V_{\mathrm{d}_k}} \leftarrow \mathrm{Dijkstra}(m_k, v_{\mathrm{d}_k})$;
$k-=1$;
end
if *k is odd number* **then**
$p_2 = p_{m_1 v_{\mathrm{s}_1}} + p_{m_1 v_{\mathrm{d}_1}}$;
$p_3 = p_{m_2 v_{\mathrm{s}_2}} + p_{m_2 v_{\mathrm{d}_2}}$;
...
$p_k = p_{m_{k-1} v_{\mathrm{s}_{k-1}}} + p_{m_{k-1} V_{\mathrm{d}_{k-1}}}$;
else
$p_1 = p_{m_1 v_{\mathrm{s}_1}} + p_{m_1 v_{\mathrm{d}_1}}$;
$p_2 = p_{m_2 v_{\mathrm{s}_2}} + p_{m_2 v_{\mathrm{d}_2}}$;
...
$p_k = p_{m_k v_{\mathrm{s}_k}} + p_{m_k v_{\mathrm{d}_k}}$;
end
return $(P_1, P_2, ..., P_k)$
end

Algorithm 1: Iterative waypoint shortest path heuristic

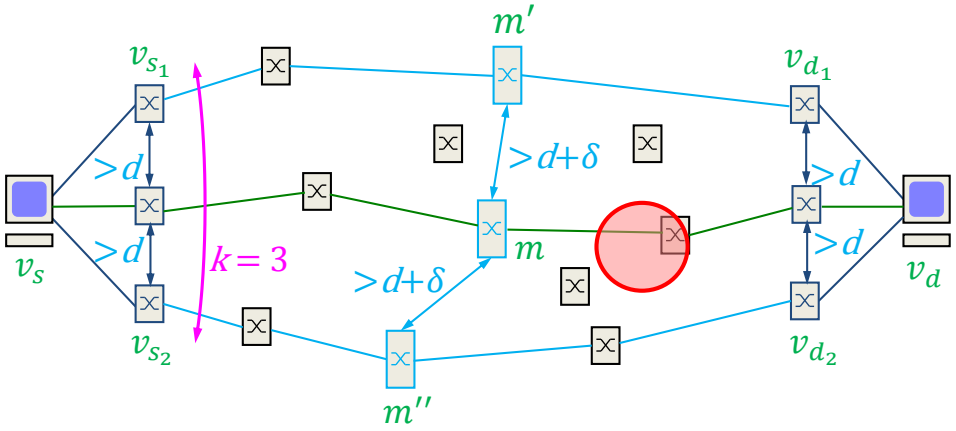

Figure 4.1: Iterative waypoint shortest path heuristic

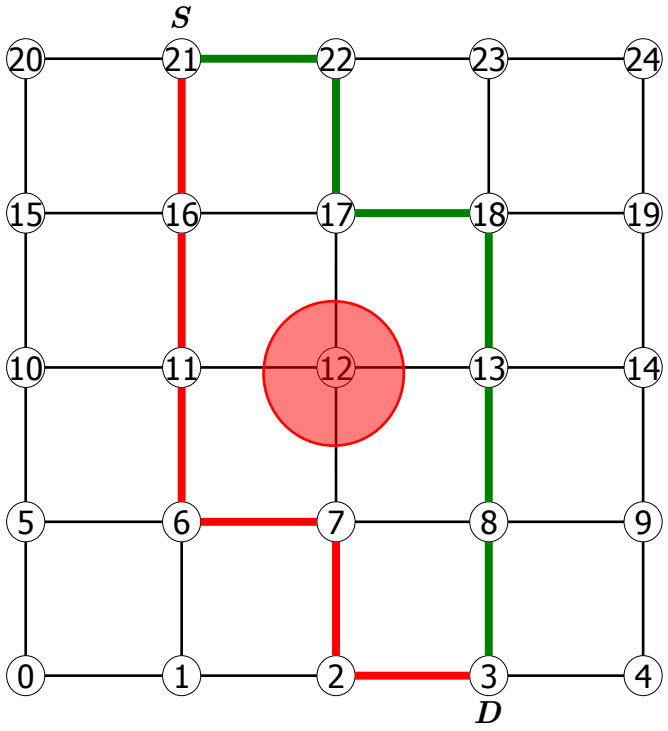

Figure 4.2: iWPSP heuristic in grid network

The second heuristic MLW is proposed and lead by UMKC [164]. MLW statistically modifies the link weights and performs Dijkstra's algorithm to calculate the GeoPaths using the modified link weights. The heuristic begins by increasing, linearly or squarely, the weight in one direction based on the perpendicular distance to the segment S con-

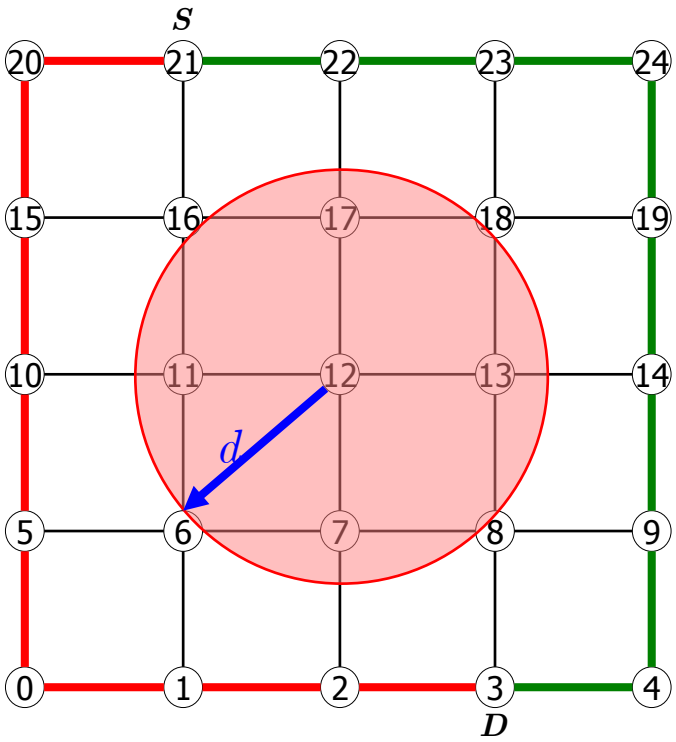

Figure 4.3: iWPSP heuristic in grid network with wider failure

necting source node v_s and destination node v_d. The weight-increment ratio is inversely proportional to the distance from the segment S. Dijkstra's algorithm is applied on the graph using the modified link weights. The heuristic repeats the process for the other perpendicular direction to S. This way the heuristic generates two GeoPaths in different directions from the shortest path.

A similar 5×5 grid network is used to demonstrate GeoPaths calculated by the MLW heuristic. As shown in Figure 4.4, MLW calculates two GeoPaths that are separated by distance d by statistically modifying link weights. Similarly, node 21 is the traffic source and node 3 is the destination. The d value is set at twice the length of the link in the grid. The weight shown in different colors is used for calculating paths in its representative color. For example, when MLW is calculating the path shown in blue-solid links (the first of the two weights before the slash), the link weight is decreasing towards the top right

corner of the grid network. The other path shown is the red-dashed links, corresponding to the second of the two weights after the slash. The detailed heuristic is proposed by UMKC and presented in Algorithm 2 for easy reference. Their simulation result is presented in [164] and we implement their heuristic in *ns-3* for performance comparison.

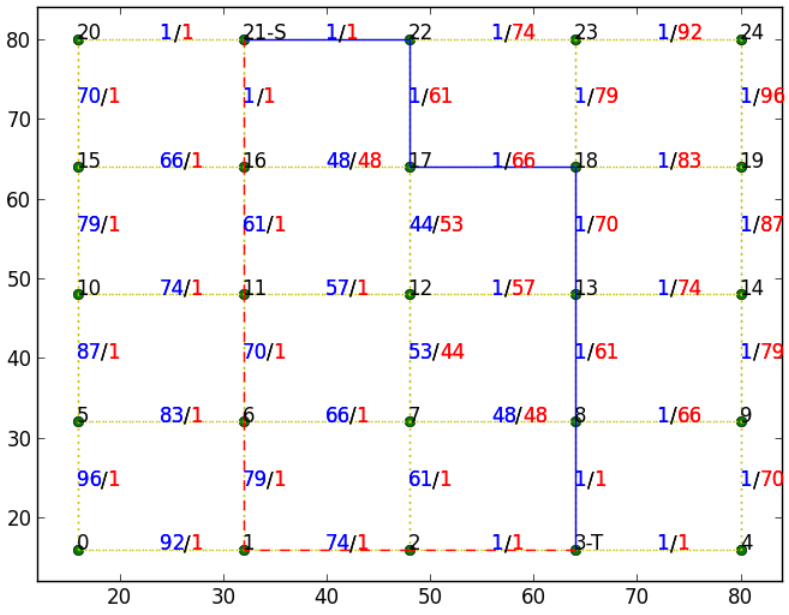

Figure 4.4: MLW heuristic in grid network

Both of the heuristics have incorporated improvement mechanisms. When the calculated paths fail to qualify the distance d-separation criteria, iWPSP chooses another waypoint with a slightly larger δ; while MLW increases the link weight around the avoidance segment. After that, both of the heuristics initialize another iteration of Dijkstra's algorithm. If the result still does not qualify the criteria, both heuristics fall back to the two-step optimal algorithm, which ensures that GeoPaths are acquired for each node pair. Another major component of the heuristics is loop detection. For example, the iWPSP heuristic can create routing loops when calculating paths for corner nodes in the topology. We use a loop detection algorithm so that if a node is visited in the calculation

Functions:
k number of GeoPaths d
begin
straight line S connecting source v_s and destination v_d
if *k is odd number* **then**
$p_1 = \text{SourceTree}_{v_d v_s} \leftarrow \text{Dijkstra}(v_d, v_s)$;
modify link weight linearly or squarely on one direction perpendicular to line S until distance d;
$p_2 = \text{SourceTree}_{v_d v_s} \leftarrow \text{Dijkstra}(v_d, v_s)$;
repeat the process for the other direction;
buffer $= d$;
$k- = 3$;
else
modify link weight linearly or squarely on one direction perpendicular to line S until distance $d/2$;
$p_1 = \text{SourceTree}_{v_d v_s} \leftarrow \text{Dijkstra}(v_d, v_s)$;
repeat the process for the other direction;
buffer $= d/2$;
$k- = 2$;
end
while $k > 0$ **do**
buffer $+= d$;
modify link weight linearly decreasing on one direction perpendicular to line S until buffer;
links beyond distance buffer, link weight = 1;
$p_{k-1} = \text{SourceTree}_{v_d v_s} \leftarrow \text{Dijkstra}(v_d, v_s)$;
repeat the process in the other direction;
$p_k = \text{SourceTree}_{v_d v_s} \leftarrow \text{Dijkstra}(v_d, v_s)$;
$k- = 1$;
end
return $(p_1, p_2, ..., p_k)$
end

Algorithm 2: Modified link weight shortest path heuristic lead by UMKC

for the same path, our heuristic detects a loop and deletes the duplicate part.

We implement both of our heuristics in *ns-3* [27] and incorporate them in our resilient routing protocol, GeoPath Diverse Routing Protocol (GeoDivRP). We base our implementation on a link-state routing protocol methodology; link-state advertisements (LSAs) are flooded throughout the network and all nodes compute their paths based on the updated topology map. GeoDivRP calculates and selects a single or multiple GeoPaths to meet the requirements from upper network layers.

4.1.3 Complexity Analysis and Evaluation

We analyze the complexity of the heuristics compared to the two-step optimal algorithm, GeoResLSR. For simplicity, we examine the complexity for obtaining two GeoPaths besides the shortest path. Since the Dijkstra's algorithm is applied n times for generating the candidate paths before selecting the qualified ones, its time complexity for generating n link-disjoint paths is $\mathrm{O}(n|V|(|E| + |V|\log|V|))$ [163]. Furthermore, The optimal algorithm demands a choice of paths that qualify the distance-separation criteria. This process requires $|V|^2$ time, which means the total complexity for the optimal algorithm is $|E| + |V|\log|V| + n|V|^2$, or $O(n|V|^2)$. The number of link-disjoint paths n is usually large to guarantee the quality of the paths calculated. For most application scenarios, n is chosen to be 1000 [165]. Therefore, for a network with nodes less than 1000, the complexity of the optimal algorithm goes up to $O(|V|^3)$.

iWPSP has a complexity of $2s^2|V|^2\log|V|$, where s is the average number of neighbors for nodes; the complexity for choosing the waypoint node is $O(|V|)$, where $|V|$ represents the number of nodes; and $2|V|\log|V|$ is for Dijkstra's algorithm to calculate two shortest paths. Therefore, the worst-case scenario is $O(n^2\log n)$. Most of the physical topologies have an average degree below four [24]. This means that s in our complexity analysis is a

small constant. This reduces the best-case time complexity for iWPSP to $O(|V| \log |V|)$. The complexity of MLW is $O(2|V| \log |V|)$, which is the complexity for invoking the Dijkstra's algorithm twice. The complexity for both of our heuristics is much better than that of the optimal algorithm, which is $O(|V|^3)$.

4.2 Real-World Network Results

In this section, we present the GeoDivRP simulation performance in *ns-3* [27] using real-world networks. We evaluate our proposed heuristics and compare their performance with the optimal algorithm when carrying UDP traffic. At the beginning of the simulation, by obtaining node locations from the link-state update messages, our protocol calculates the GeoPaths. When the simulation begins, our protocol starts sending data traffic using the shortest path. When a challenge occurs in the network, GeoDivRP responds to the failure faster than Open Shortest Path First (OSPF) [166] and use the pre-calculated paths according to the challenge estimation. We have incorporated a fallback mechanism; when the generated GeoPaths do not satisfy the application requirement, OSPF is used for further routing decisions. Before introducing the simulation result in the Internet Service Providers (ISPs) network, we provide verification results for the heuristics.

4.2.1 Routing Heuristic Verification

We present the GeoPaths calculated by our heuristics using the Nobel-EU (Pan-European Reference Network) with 28 nodes and 40 links [52]. We assume a challenge along the line from Amsterdam to Rome with a radius of 50 km. Nodes Strasbourg and Frankfurt are in the challenge circle. The result of iWPSP is shown in Figure 4.5 with the challenge region shown in a red circle. We show the paths calculated for all the node pairs in solid lines. And the two paths provided to the node pair of Amsterdam – Rome

are Amsterdam–Hamburg–Berlin–Munich–Vienna–Zagreb–Rome for the first path, and Amsterdam–Brussels–Paris–Lyon–Zurich–Millan–Rome for the second one.

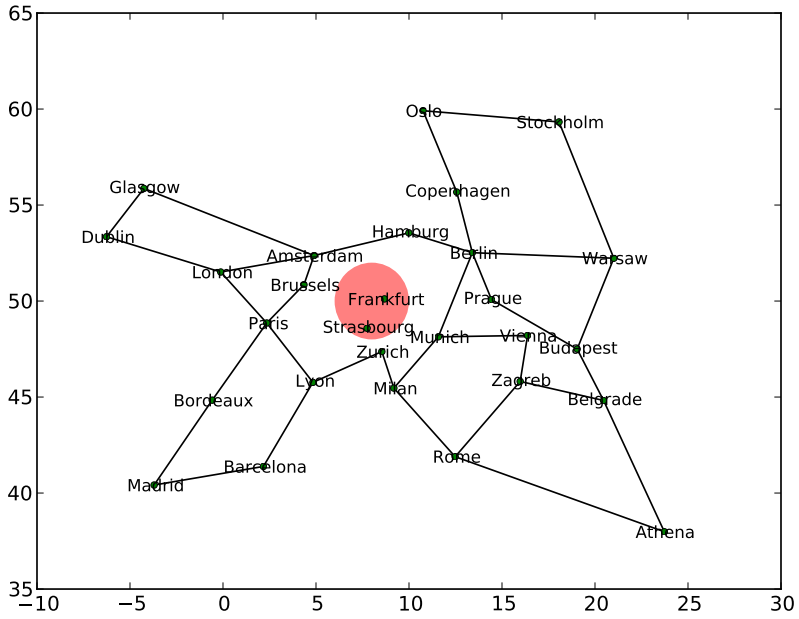

Figure 4.5: iWPSP heuristic in Nobel-EU network

The result of MLW using the same challenge radius is shown in Figure 4.6 with its two paths from Amsterdam to Rome. The first path shown in red-dashed link is Amsterdam–Hamburg–Berlin–Munich–Vienna–Zagreb–Rome, and the second path shown in blue solid link is Amsterdam–Brussels–Paris–Lyon–Rome. The first path is exactly the same with iWPSP, while the second one for MLW avoids the Lyon–Zurich link. We present a large radius challenge case when using MLW in Figure 4.7. The two paths shown in red and blue are further apart to bypass the challenge.

We present the execution time of the heuristics to demonstrate their effectiveness compared to the optimal algorithm in the case of calculating two GeoPaths. The evaluation is performed on a Linux machine with 3.16GHz Core 2 Duo CPU with 4GB memory. We use an increasing dimension of grid networks to analyze the time complexity. The

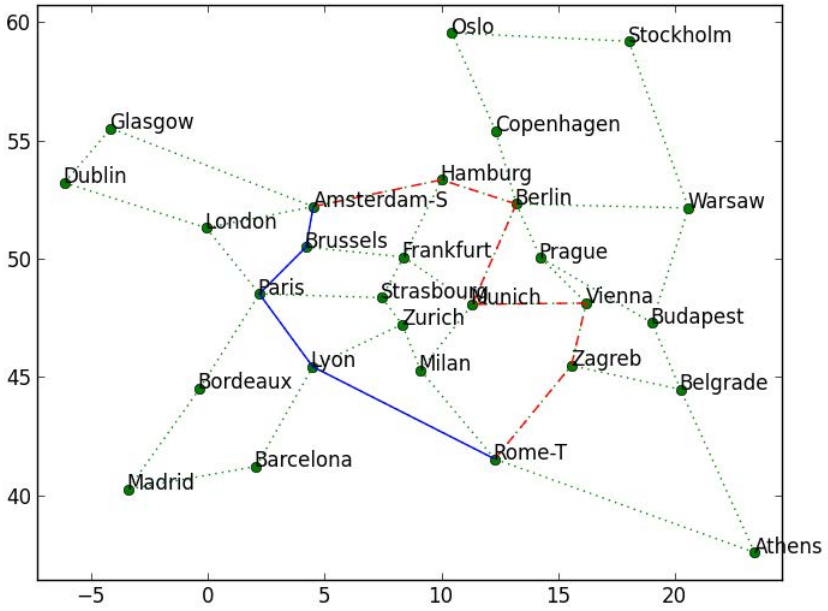

Figure 4.6: MLW heuristic by UMKC in Nobel-EU network

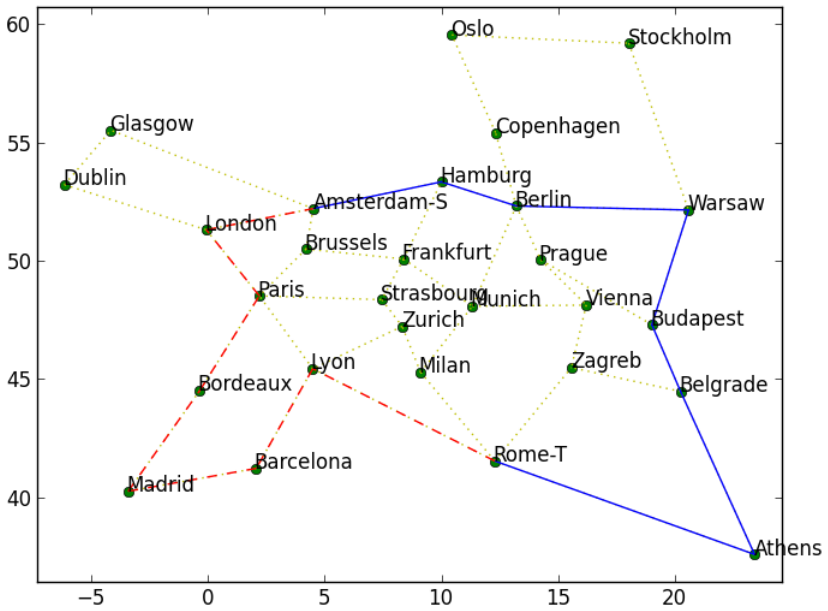

Figure 4.7: MLW heuristic by UMCK in Nobel-EU network with large radius

grid dimension ranges from 3×3 to 11×11; meaning that the number of nodes varies from 9 to 121. We present the time to calculate GeoPaths for all the node pairs in the topology. Note that when calculating only one node pair that happens more often in real-world scenarios, the time is exponentially less. As shown in Figure 4.8, the x-axis is the grid dimension and the y-axis is the log-level algorithm execution time in seconds. Both MLW and iWPSP algorithms show better execution time compared to the two-step optimal algorithm. For example, when calculating all the paths in 11×11 grid, MLW and iWPSP take 20 s and 65 s respectively, while the optimal algorithm takes greater than 3000 s. Furthermore, we observe that iWPSP has a greater execution time compared to that of MLW. This is because of the one extra iteration of the Dijkstra's algorithm and the selection of qualifying waypoint nodes. However, we observe that when calculating GeoPaths in real-world topologies, iWPSP is more efficient in calculating the paths for node pair around the topology boundary. This is because by selecting waypoints based on a distance and a delta value, iWPSP has more control over the distance separated from the two paths. A better algorithm might be combining the two heuristics in calculating a single topology, and this is planned in future work.

4.2.2 Routing Performance

We now present simulation results using physical topologies including Sprint [24], Level 3 [151], Internet2 [158], and TeliaSonera [159]. We use constant bit rate (CBR) UDP traffic, sending from each node to all the others at a data rate of 1 packet/s, with a 1000 byte packet size. The link bandwidth is 10 Mb/s and the delay is 2 ms. We carry out the simulation once for each topology since there is no randomness because of the CBR traffic. There are three deterministic geo-correlated challenges we have simulated. From 20 to 40 s, the challenge occurs around Los Angeles, from 60 to 80 s in Kansas City, and the last challenge occurs in New York City from 100 to 120 s. The challenge locations come

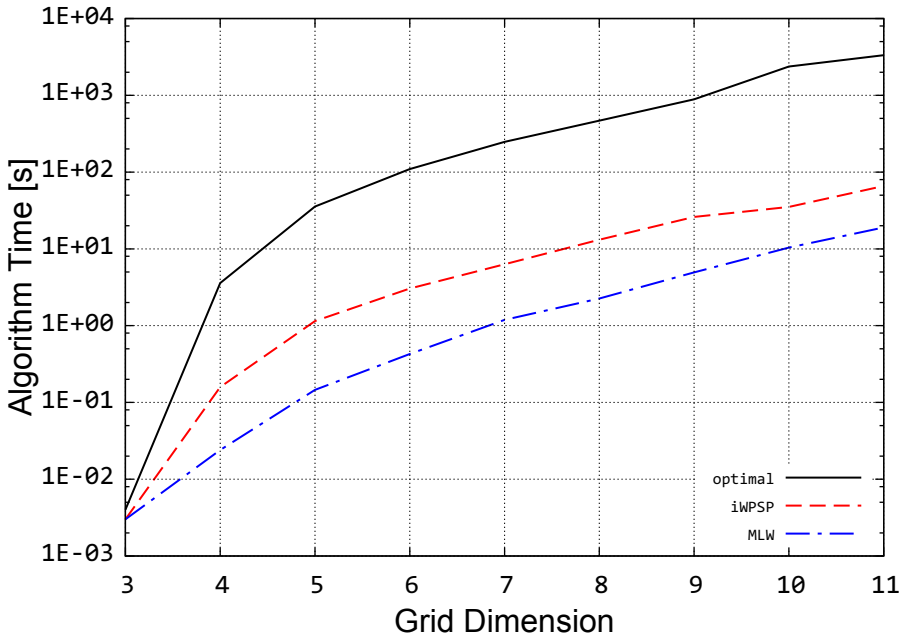

Figure 4.8: Heuristics complexity analysis and comparison

from the flow robustness analysis in the previous chapter, and our challenge duration time is set as 20 s. We choose these different challenge areas so that the most vulnerable area is around Kansas City, due to its high betweenness as a major fiber exchange point in the US. The next damage area is around New York City. While it does not have many high-betweenness nodes, the network is dense and more nodes are challenged in a given radius. The least vulnerable area is around Los Angeles. The radii of the three challenge areas are 300 km. By assuming the correct estimation of the challenge radius and position, we compare our protocol's performance with standard OSPF in terms of the packet delivery ratio (PDR) as well as delay. PDR is the ratio of packets delivered divided by total packets sent, while delay is the time it takes for the data packet to travel end-to-end. We use the same challenge areas throughout all the topologies in this chapter for ease of comparison. The iWPSP heuristic is used in GeoDivRP for calculating the GeoPaths. MLW [2] achieves the same PDR and delay result as iWPSP when the links

[2]Proposed and lead by UMKC

are carefully modified to guarantee the distance-separation criteria. Since *ns-3* is an event-driven network simulator and the algorithm execution time is not included in the simulation time, the delay in *ns-3* for both the iWPSP and MLW is the same.

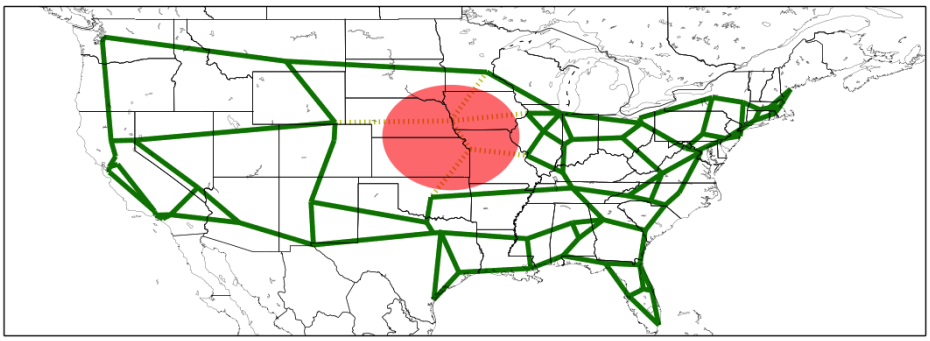

Figure 4.9: Sprint topology under regional challenges

The Sprint physical network contains 77 nodes and 114 links and the GeoPaths calculated by GeoDivRP to bypass the challenge is shown in Figure 4.9. The red circle shown in this figure is the challenge area at Kansas City. The PDR result for the Sprint network is shown in Figure 4.10. We compare the performance of our GeoDivRP with standard OSPF. The second challenge at Kansas City area occurs at 60 s and GeoDivRP shows substantial performance improvement compared to OSPF. The PDR of OSPF drops to 75% and it takes 10 s to converge while the time for GeoDivRP is within 1 s and the PDR only drops by 2%. The last challenge occurs from 100 s to 120 s and the difference in PDR between OSPF and GeoDivRP is small, only about 1%. This is because the challenge at New York City has little effect on the connectivity of the overall topology. The PDR for OSPF drops by about 1% and it takes 10 s to recover, and there is no noticeable PDR drop for our protocol. The first challenge happens at 20 s to 40 s and there is no noticeable PDR drop for both of the protocols. This is due to the same reason as in New York City but the loss of PDR for both GeoDivRP and OSPF is even less.

The delay analysis for Sprint network is shown in Figure 4.11. The reason that OSPF

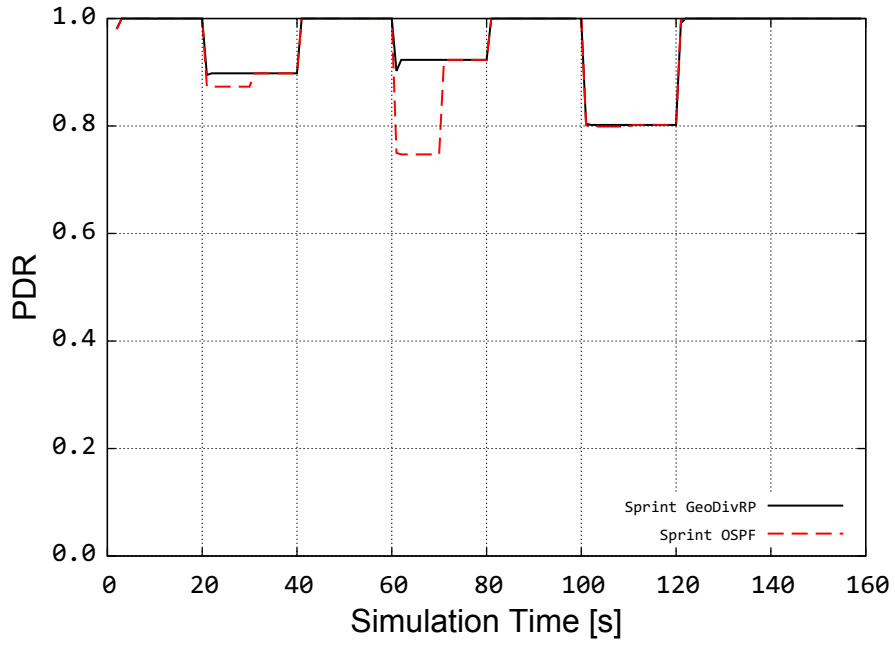

Figure 4.10: Sprint PDR under regional challenges

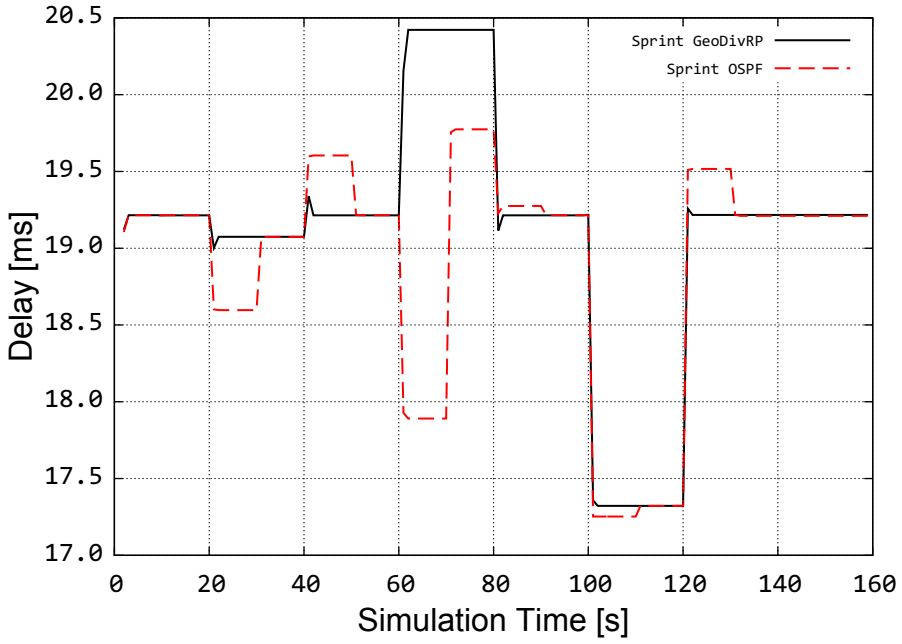

Figure 4.11: Sprint delay under regional challenges

shows lower delay is because most of the data packets during the challenge have been dropped and the lost packets are not counted as delay. This is also the reason that there is a delay drop for OSPF before converging. Consider the first challenge in Figure 4.11, the delay for OSPF drops from 20 – 30 s due to the packet losses, while GeoDivRP converges and calculates geodiverse paths during that period of time and shows 1 s higher in delay. However, the extra delay is caused by extra path stretch due to routing packets around the failed region. We also notice a delay bump for OSPF right after the challenge is finished. For example, in Figure 4.11, from 40 – 50 s, there is an increase in delay for OSPF. The same occurs at 80 – 90 s, and 120 – 130 s. This is because OSPF needs to reconverge after the topology has recovered from the challenge. In contrast, for our protocol, the extra convergence time is still 1 s and no noticeable delay increase is recorded.

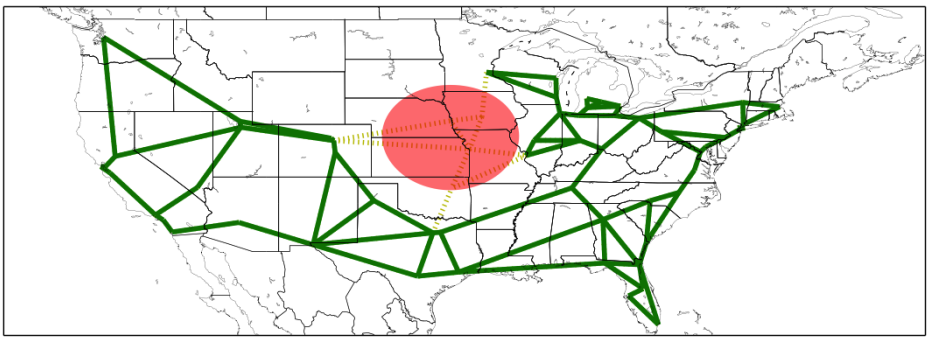

Figure 4.12: Level 3 topology under regional challenges

The Level 3 physical network contains 99 nodes and 132 links. As shown in Figure 4.12, the similar challenge location as from the Sprint network has caused more nodes and links to fail. The PDR for the Level 3 network is shown in Figure 4.13. Since Level 3 shares geographical similarities with the Sprint network; we observe a similar PDR result. The challenge in the Kansas City area reduces the PDR for OSPF significantly; it is even greater than for Sprint. This is because the Level 3 network lacks some of the nodes and

links from Seattle to Chicago and the challenge around the Kansas City area causes more damage to the overall connectivity. The delay case for the Level 3 network is similar to the Sprint network as shown in Figure 4.14.

The Internet2 physical network is a smaller research network with only 16 nodes and 24 links. The PDR for the Internet2 network is shown in Figure 4.15. The challenged PDR and delay show a similar trend as previous topologies. The first challenge does damage to the network connectivity and GeoDivRP converges within 1 s. The second challenge in Kansas City area causes OSPF to drop around 10% in the PDR and takes 10 s to converge and return the PDR to normal. The Los Angeles challenge has small impact on the network similar to the Sprint case. The delay analysis for the Internet2 network is shown in Figure 4.16. For the same reason, OSPF shows a smaller delay compared to that of the GeoDivRP during challenges from 20 – 30 s, 60 – 70 s, and 100 – 110 s.

The TeliaSonera physical network contains 18 nodes and 21 links. The PDR for TeliaSonera is shown in Figure 4.17. The second challenge at Kansas City area drops the PDR for OSPF to around 50%. This significant drop is caused by two reasons. First, the Kansas City node connects multiple nodes between the east and west coast. Second, the TeliaSonera network is very sparse so the damage from the Kansas City node is greater than that for the other networks. However, GeoDivRP recovers from the damage in only 1 s and limits the PDR drop within 1%. The PDR case for both the first and the third challenge are similar. At the same time, OSPF drops about 1% of the total packets and recovers only after 10 s. The delay analysis is shown in Figure 4.18. OSPF shows a smaller delay during challenges since the dropped packets are not counted for delay analysis. We notice that the delay increases after the challenge for OSPF at 80–90 s is larger than other challenge locations as well as the same challenge location in other topologies. This is because OSPF is using a path with more path stretch before convergence.

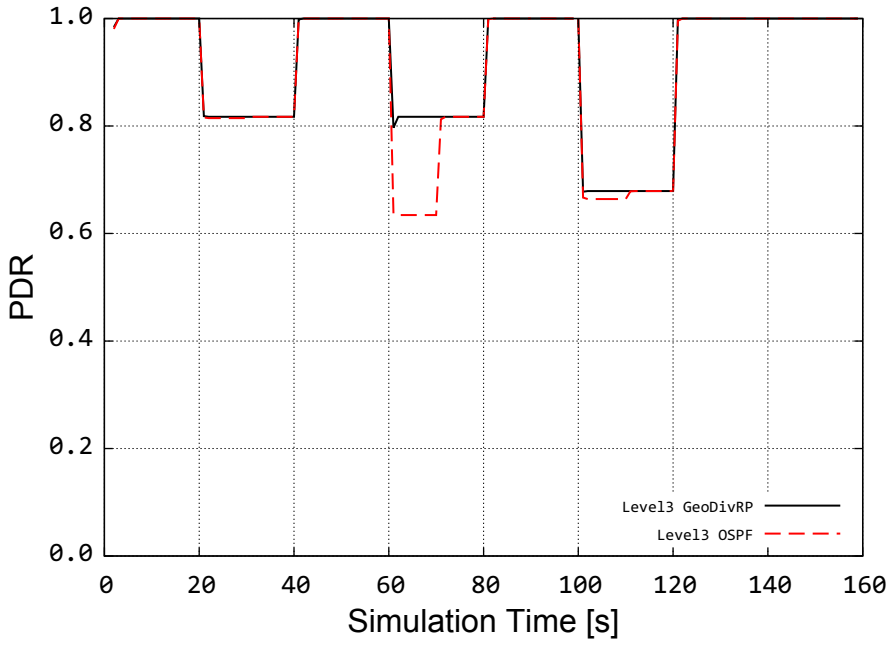

Figure 4.13: Level 3 PDR under regional challenges

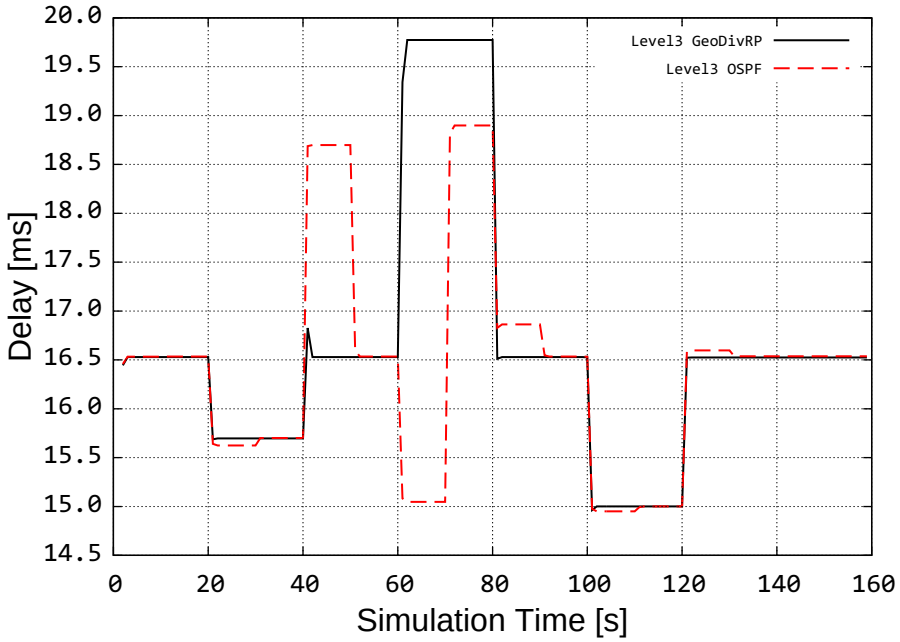

Figure 4.14: Level 3 network delay under regional challenges

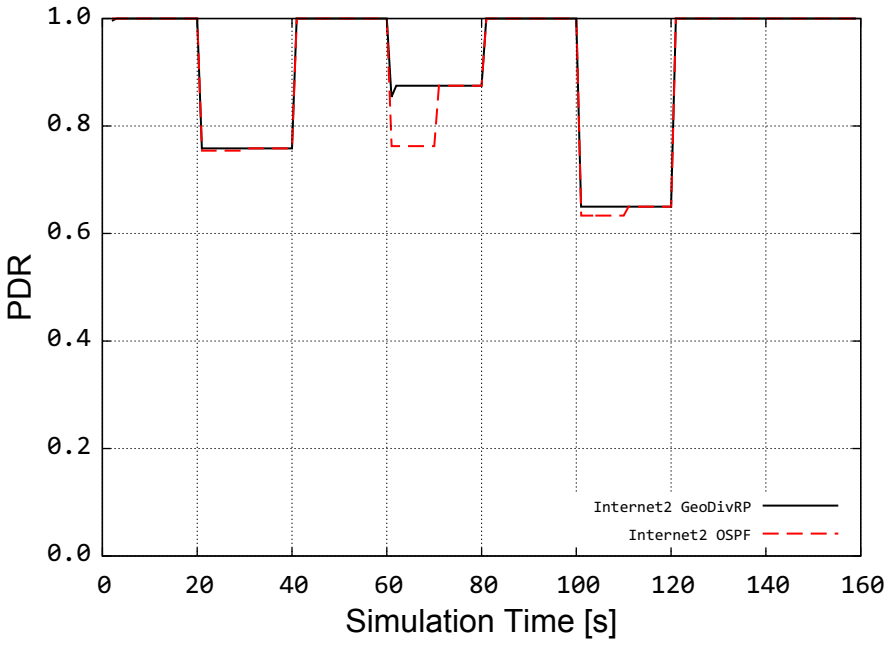

Figure 4.15: Internet2 PDR under regional challenges

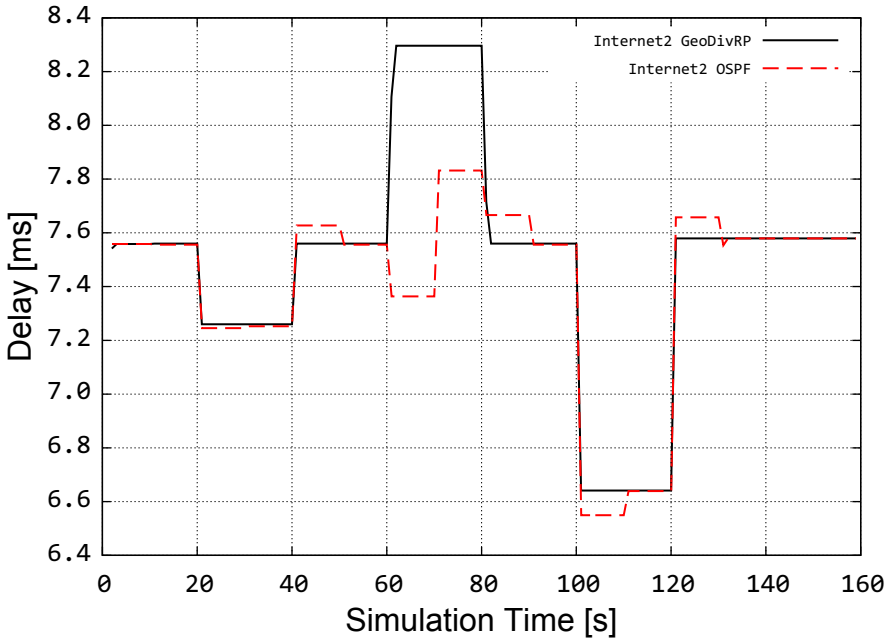

Figure 4.16: Internet2 delay under regional challenges

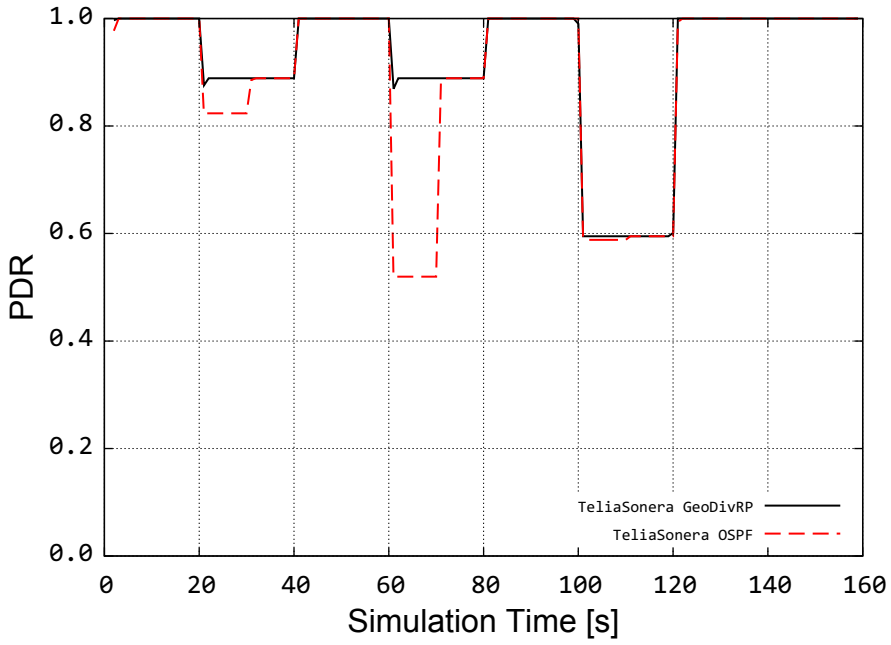

Figure 4.17: TeliaSonera PDR under regional challenges

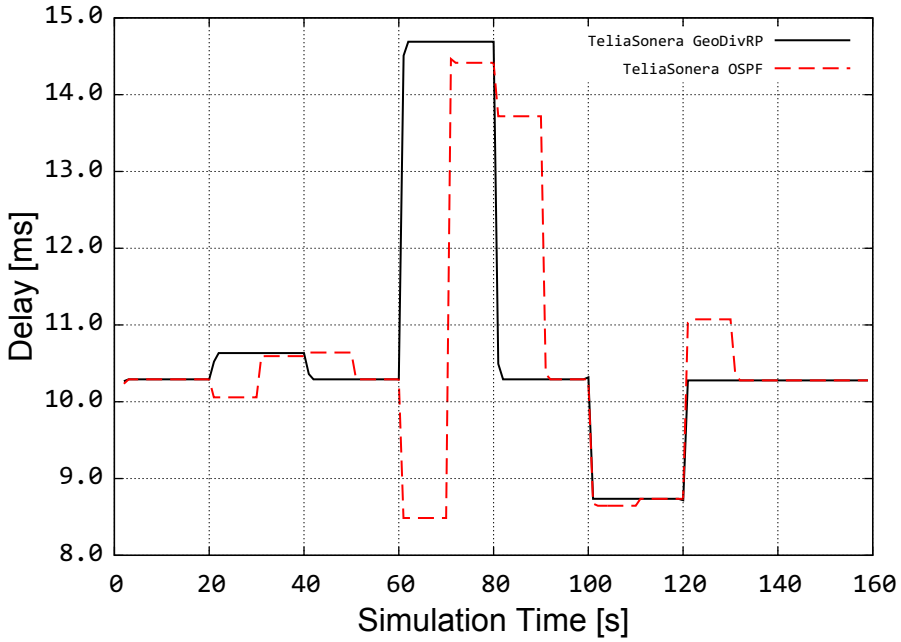

Figure 4.18: TeliaSonera delay under regional challenges

We argue that GeoDivRP performs well in the face of geo-correlated challenges. First, the iWPSP routing heuristic returns GeoPaths with controlled algorithm and time complexity. Second, it presents improved packet delivery ratio (PDR) and small delay increase when compared to OSPF.

4.2.3 Disaster Mitigation

Based on the critical regions identified using the mechanism introduce in Chapter 3, various protection mechanisms can be applied using GeoDivRP. We perform a disaster mitigation analysis for the flow robustness target equals 0.6 from Table 4.2. The same table has been shown in Chapter 3, it is shown here again for easier reference. By restoring failed nodes one by one beginning with the highest betweenness centrality, the flow robustness improvement is significant. The reason for adding nodes with higher betweenness centrality is that betweenness defines the number of shortest path passing through a node and can offer better restoration results with traffic considered. As shown in Figure 4.19, with only two protected nodes, the flow robustness for all the topologies increases from below 50% to around 80%.

Network Challenge Simulation

We further carry out network simulation to evaluate the mitigation results and demonstrate the performance of GeoDivRP. *ns-3* [27] is used with the link bandwidth as 10 Mb/s and the delay as 2 ms, similar to the previous simulation setup. The total simulation time is 100 s and two challenges are introduced; the first challenge starts from 20 to 40 s. The second challenge occurs from 60 to 80 s with the protected nodes. The protected nodes come from the challenged node set identified for the flow robustness target equals 0.6 as shown in Table 4.2. The total protected nodes are three out of the six failed ones,

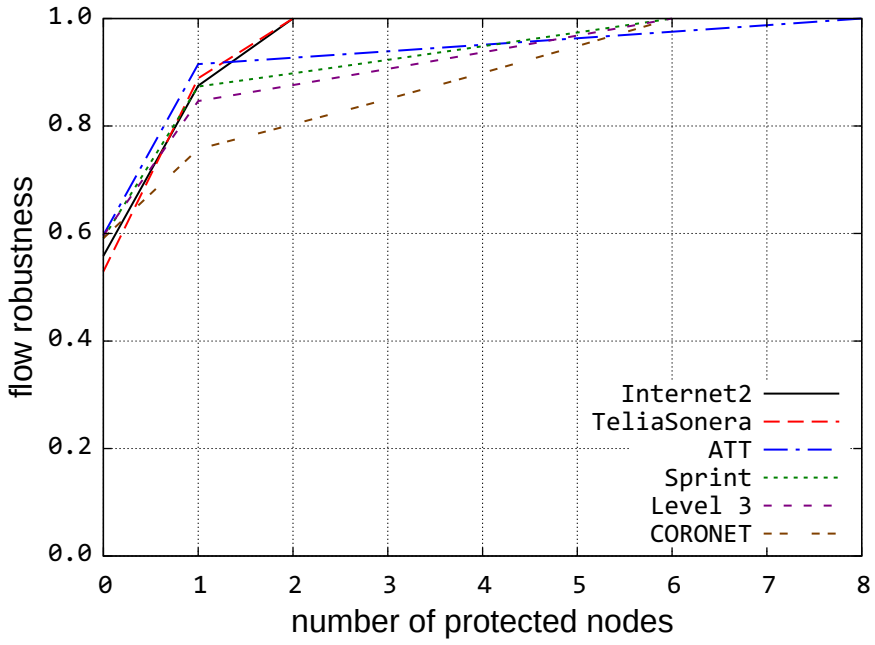

Figure 4.19: Flow robustness improvement for unweighted graph

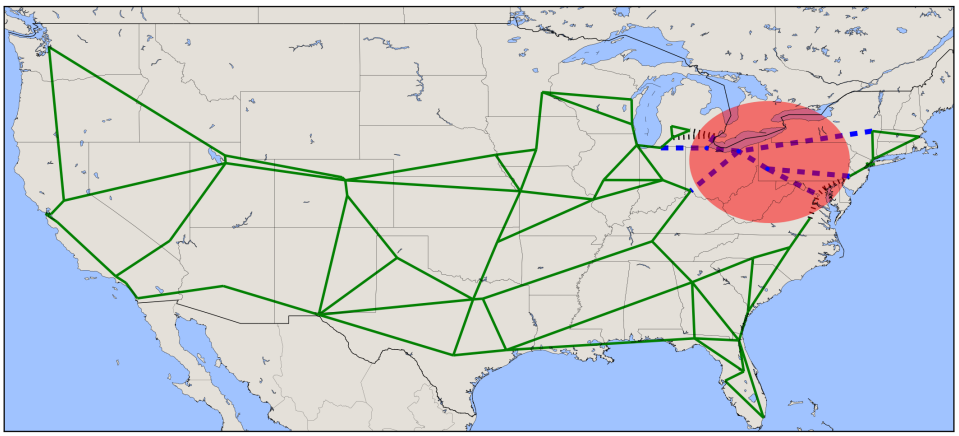

Figure 4.20: Level 3 network challenge location with protected nodes

which means three of the highest betweenness nodes are added to the failed topology during the second challenge.

We present the improvement result for the Level 3 network. As shown in Figure 4.20, the challenge location is for the flow robustness target equals 0.6. Nodes in the range of the circle are disrupted, along with the links connecting to them as shown in black

dotted lines. The three protected nodes and the adjacent links are shown in blue dashed lines.

As shown in Figure 4.21, for the first unprotected challenge, the PDR drops to around 60%, which closely matches the flow robustness result. OSPF needs 10 s to converge after the challenge, which is shown as the PDR decreases from 20 to 30 s. On the other hand, it takes only 1 s for GeoDivRP to reconverge and provide paths bypassing the challenge. The second challenge with the protected nodes has a PDR above 90%. For the same reason, it takes OSPF 10 s to converge and the PDR decrease is larger compared to the previous challenge; with the protected nodes, some previously disconnected nodes are connected during this challenge and OSPF cannot provide shortest path for the newly connected node pairs until reconvergence.

As shown in Figure 4.22, the end-to-end delay for OSPF drops during the challenge before reconvergence because OSPF has around 5% to 10% (first and second challenge respectively) more packet drops compared to GeoDivRP and the dropped packets are not counted in the delay result. After the convergence, from 30 to 40 s and 70 to 80 s, there is 1 s extra delay for GeoDivRP compared to OSPF. This is because GeoDivRP calculates paths with greater path stretch provided by the routing heuristic. However, 1 s extra delay is justified by the 5% to 10% PDR improvement.

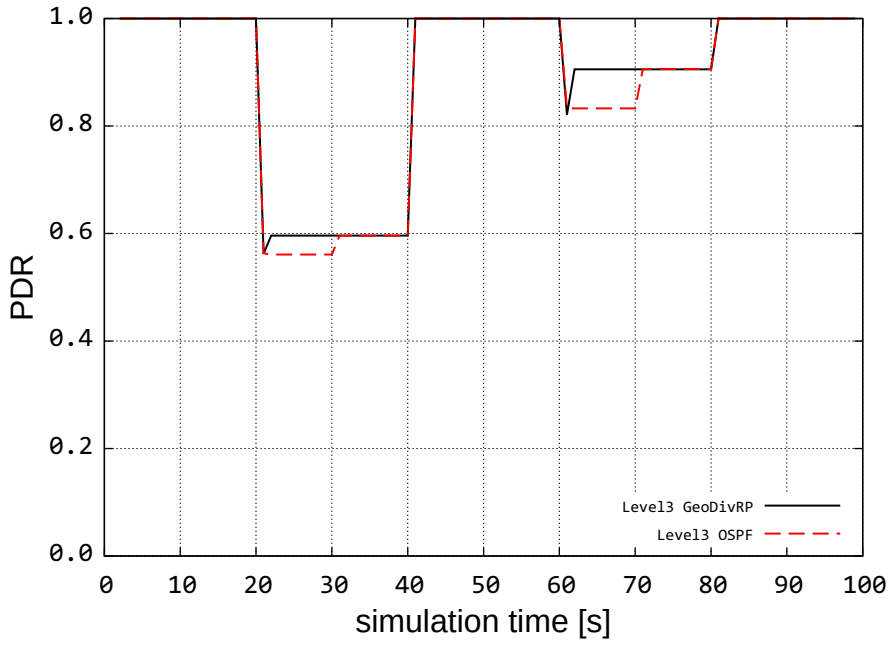

Figure 4.21: Level 3 network packet delivery ratio

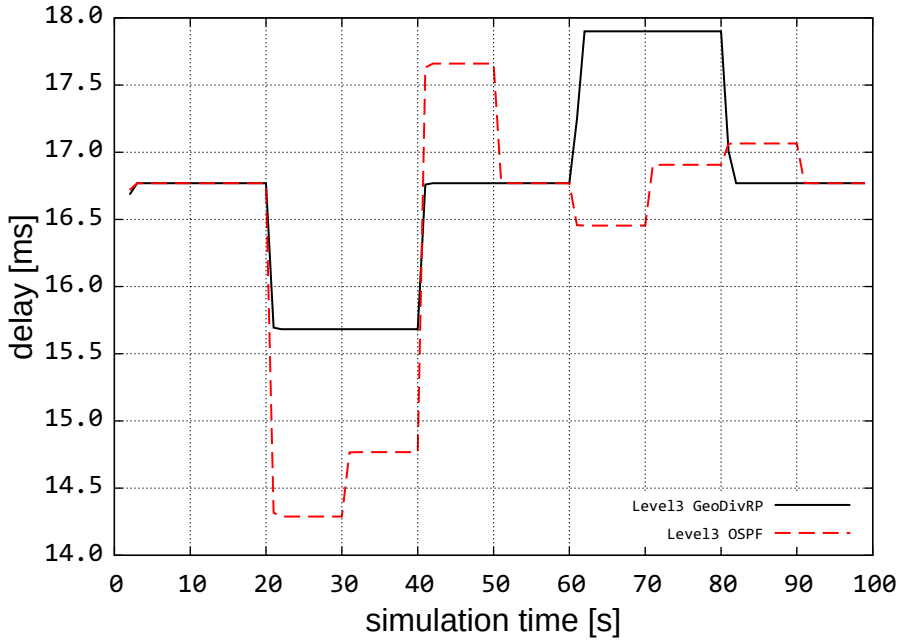

Figure 4.22: Level 3 network end-to-end delay

Table 4.2: Physical topology vulnerable locations (FR=0.6)

Network	Number of Nodes	Number of Links	Flow Robustness	Challenge Centers	Challenge Coordinates	Challenge Radius (Km)	Number of Failed Nodes
AT&T	162	244	0.59	Morgantown, WV	39.67, -79.81	256	8
CORONET	39	63	0.59	West Decatur, PA	40.95, -78.32	289	6
Internet2	16	24	0.56	Stahlstown, PA	40.19, -79.35	246	2
Level 3	63	94	0.59	Butler, PA	40.84, -79.86	325	6
Sprint	77	114	0.59	Rockwood, PA	39.99, -79.27	228	6
TeliaSonera	18	21	0.53	Greensburg, PA	40.26, -79.58	225	2

Chapter 5

Flow Geodiverse Problem

When multiple paths are provided to each network device, how the traffic is distributed among them requires comprehensive study. In this chapter, we propose the *flow geodiverse problem* (FGD) and two optimization formulations to solve it. We begin our discussion by proposing a novel cross-layer protocol stack, ResTP–GeoDivRP, in Section 5.1. The protocol stack benefits from the cross-layer communication and provides multiple GeoPaths to network devices for resilient data transmission. In Section 5.2, we present the flow-geodiverse optimization design for GeoDivRP to satisfy either the minimum-cost or the path delay-skew requirement passed from ResTP, our resilient transport protocol. We extend our GeoDivRP routing algorithm to provide the optimal traffic allocation information on multiple paths for different node pairs or commodities[1]. Finally, we present the performance of our flow-geodiverse optimization engine and our ResTP–GeoDivRP protocol stack when using the multipath forwarding mechanism in Section 5.3.

5.1 ResTP–GeoDivRP Network Stack

We propose a ResTP–GeoDivRP resilient network stack for dependable network communication through the cross-layer information. Using the GeoPaths provided by Geo-

[1]We use node pair and commodity interchangeably

DivRP, ResTP establishes multiple subflows for each node pair to achieve better performance and resilience.

5.1.1 Cross-Layer Protocol Integration

Our cross-layer network stack integrates GeoDivRP with ResTP [128, 129, 167], our resilient transport protocol [2]. GeoDivRP and ResTP fit in the protocol stack as shown in Figure 5.1. *Knobs* $\mathbb{K}$ are used by higher layers to influence the lower layer operation while *dials* $\mathbb{D}$ are the mechanisms for lower layers to provide instrumentation to the layers above. The application passes a service specification (ss) and a threat model (tm) down to ResTP. Upon receiving these parameters, ResTP determines the type of transport service needed (including error control and multipath characteristics) and requests that GeoDivRP calculates GeoPaths that meet the requirement tuple $(k, d, [h, t])$. k is the total number of GeoPaths requested, d is the distance-separation criteria, in which any two nodes on disjoint paths are separated by a distance greater than d, $[h, t]$ are the desired constraints on path stretch h (number of additional hops for diverse paths) and the temporal skew (delay difference) across paths, t. Throughout this work, k is chosen as three for two main reasons. First, the node degree for the physical topologies used in this work is below four. Second, a common spread used in erasure coding is three, which masks a single path failure. Note that GeoDivRP interprets path stretch h as path delay (latency) l. Based on the configuration and network statistics ($[l, f]$) collected from the network monitoring engine and the requirement tuple from ResTP, GeoDivRP calculates the geodiverse path set $P_k = p_0 ... p_{k-1}$. The statistic f represents the node and link failure information. It passes the configuration (P_k, l, t) to the optimization engine as shown in Figure 5.2. Based on the latency (l) and skew (t) requirement, the optimization engine returns the path set P_k along with its traffic allocation information X_k to

[2]The design and analysis of ResTP is not part of this dissertation

GeoDivRP, which are then passed up to ResTP for establishing subflows. Upon receiving the GeoPaths, ResTP establishes multiflow with error control needed to meet the service specification, including the per-subflow error control (ARQ, hybrid ARQ, FEC, or none) and flow bundle (e.g., 2-of-3 erasure code for real-time critical service or 1+1 redundancy with a hot-standby for delay and loss tolerant service). The multipath forwarding is applied in the context of several real-world service provider networks to analyze the diversity gain and improvement in terms of throughput.

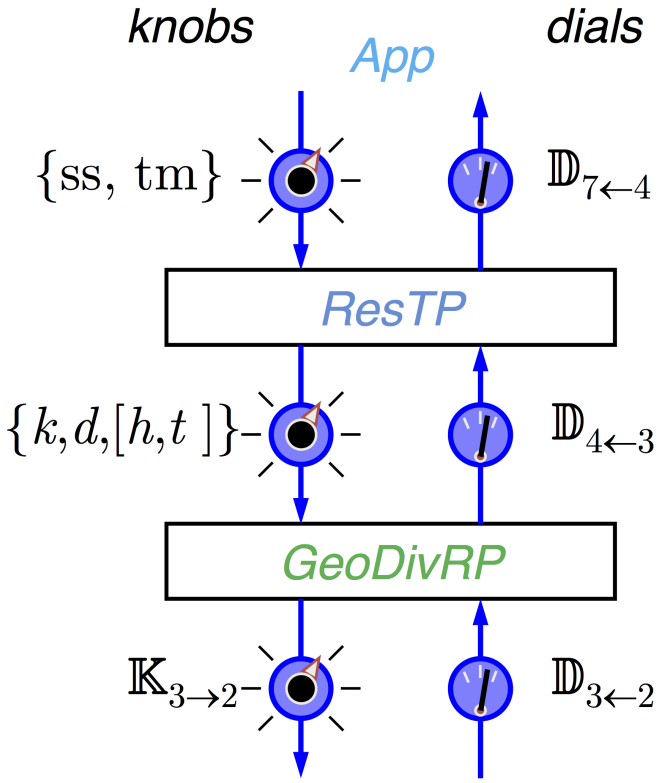

Figure 5.1: Block diagram of the GeoDivRP and ResTP

ResTP is still under active development and not complete; for this dissertation, we are only using the cross-layering feature of ResTP to take advantage of the geodiverse paths returned by GeoDivRP. Since the congestion control mechanism for ResTP is not yet implemented, in the multipath forwarding simulation, we use a scenario that the link

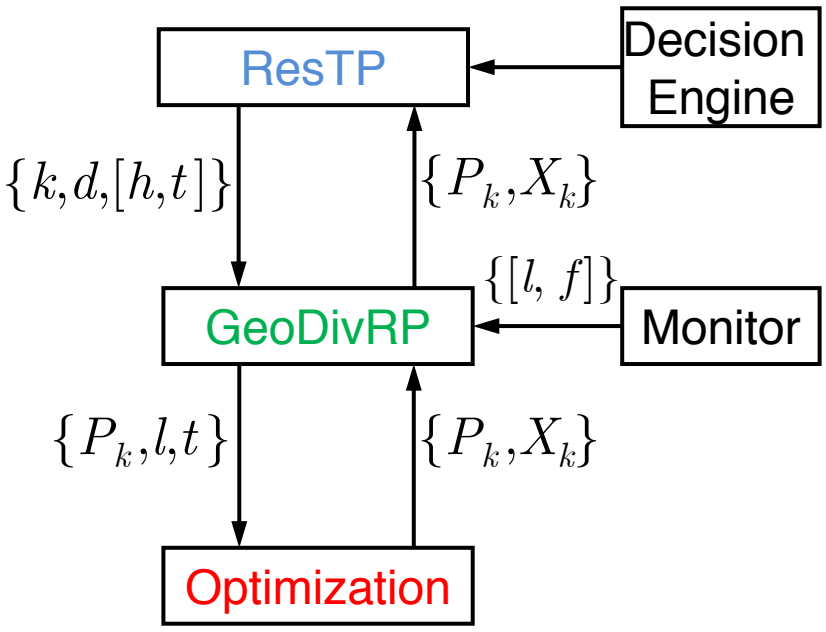

Figure 5.2: GeoDivRP and optimization engine

bandwidth is over-provisioned to avoid network congestion.

5.2 Flow Geodiverse Optimization

We design our optimization engine to provide optimized traffic information to GeoDivRP when calculating the GeoPaths. We formulate a minimum-cost routing problem using a linear programming (LP) model and a delay-skew minimization problem using a nonlinear programming (NLP) model. The paths for both of the problems are provided by a modified iWPSP routing heuristic. When the network is under geo-correlated challenges, the rerouted traffic has a limited number of backup paths to select from, which raises the potential danger for the network to get congested. The congestion will further cause higher end-to-end delay. We consider the problem of establishing multiple bounded delay-skew GeoPaths with a given demand matrix when the challenge occurs. Our model assumes weighted links and we calculate the maximum throughput that multipath rout-

ing can achieve. We have formulated both of our problems as multi-commodity flow problems.

Similar to GeoDivRP without an optimization requirement, we consider the link-state routing environment, in which each node maintains a map of the network link interconnection. We use the link-congestion factor to understand how a routing protocol utilizes network resources, which has shown to be a good indicator for network congestions [120, 168]. Link flow x_e (when $e \in E$) is defined as the total flow that has been assigned on the link e after optimization, with E representing the link (edge) set. The value x_e/u_e is the link-congestion factor and a link is overloaded if the utilization exceeds 100%. u_e is the flow upper bound on link e. The value $\max_{e \in E}(x_e/u_e)$ is the network-congestion factor; it is the maximum link utilization value over all links in the network. In this work, we allow the link-congestion factor to have values larger than 100%. The reason is that the simulation model is not using buffers for intermediate nodes, and all the extra data packets assigned to an overloaded link would be dropped. This assumption facilitates the representation of overloaded links for OSPF with shortest path routing.

5.2.1 Minimum-Cost Optimization

We start our discussion with the minimum-cost optimization problem. It targets at minimizing the overall transmission cost for all commodities without overflowing any network link. The formulation is based the link-path approach [138, 143] for multi-commodity flows. It incorporates the GeoPaths provided by GeoDivRP for each commodity.

For a commodity w, let P^w denotes the collection of all GeoPaths from the source node v_s^w to the destination node v_d^w. We use variable x_p^w as the flow on path p for commodity w. The link-path indicator variable is defined as $\eta_e^w(p)$; it is one if link(e) is contained in the path p, and is zero otherwise. We list the important variables used in the optimization

models in Table 5.1.

Table 5.1: Notation for optimization problem formulations

	Description
P^w	candidate number of paths to be considered for commodity w
c_p^w	cost per unit flow on path p for commodity w
x_p^w	traffic flow on path p for commodity w (variable)
u_e	flow upper bound on link e
q^w	demand for commodity w
$\eta_e^w(p)$	=1 if link e belongs to path p ($p \in p^w$); 0, otherwise
k^w	number of GeoPaths requested for commodity w
L	total packet delay in the network
T	overall path skew for all commodities
γ	weight parameter tuning delay and skew

The GeoDivRP linear optimization problem can now be stated as follows:

$$\min \sum_{w \in W} \sum_{p \in P^w} c_p^w x_p^w \tag{5.1}$$

subject to

$$\sum_{p \in P^w} x_p^w = q^w, \quad w \in W \tag{5.2}$$

$$\sum_{w \in W} \sum_{p \in P^w} \eta_e^w(p) x_p^w \leq u_e, \quad e \in E \tag{5.3}$$

$$x_p^w \geqslant q^w / k^w, \quad p \in P^w, \ \ w \in W. \tag{5.4}$$

The objective function shown in Equation 5.1 minimizes the overall cost of flows over different paths for all the commodities. Equation 5.2 is the flow conservation constraint over all paths $p \in P^w$ of traffic demand q^w for each commodity w. Equation 5.3 is the link capacity constraint for each link e requiring that the sum of the path flows passing through that link is at most its capacity upper bound u_e. Equation 5.4 requires all path

flow variables to be greater than or equal to a minimum path flow for traffic diversity, captured by the total traffic demand divided by the minimum number of geodiverse paths k^w to be considered for each commodity w. Note that $k^w \leq \#(P^w)$ and typically, $k^w < \#(P^w)$ (otherwise, the flow will be equally distributed along all the paths for a commodity). Clearly, Equation 5.4 forces multipath flow, an important requirement for our GeoDivRP approach.

All the candidate paths provided to the optimization problems are geodiverse and they are provided by our modified iWPSP routing heuristic shown in Algorithm 3. The paths returned from iWPSP are all simple paths, with δ controlling the skew result for different geodiverse paths. The skew constraint t is passed down along with the other parameters from ResTP. This heuristic naturally controls the skew for different paths in different commodities using the distance-separation criteria δ. By slightly increasing or decreasing the δ value along each direction during the path calculation, we can indirectly alter the skew value of the returned GeoPaths. If the returned path set is not bounded by the provided skew requirement, iWPSP uses a different δ value to obtain another set.

5.2.2 Delay-Skew Optimization

For different applications, the requirement for the path delay or skew varies. For example, data traffic is more sensitive to delay while multimedia traffic is more so to skew. The minimum-cost optimization provides the optimal traffic allocation ratio on the GeoPaths while minimizing the overall network cost. However, it does not have a direct control over the path delay or skew and merit the requirement passed from ResTP. Thus, we propose another formulation that considers both path delay and skew as an weighted objective. It provides a flexible way to manage the emphasis on either delay or skew depending on the application scenario. GeoDivRP calculates the GeoPaths that satisfies the delay-skew requirement using the non-linear optimization algorithm if permitted. Otherwise,

Functions:
Calculate k number of d-distance separated skew-bounded paths
begin
segment S connecting source v_{s} and destination v_{d}, with its middle point m;
choose neighbor node v_{s_k}, v_{d_k} at least d distance from $v_{\mathrm{s}_{k-1}}$, $v_{\mathrm{d}_{k-1}}$, respectively;
if *k is odd* **then**
choose two nodes m_1 and m_2 that are separated by $d+\delta$ on each direction of S, where $m_1 m m_2$ is perpendicular bisector of S;
$p_1 = \mathrm{SourceTree}_{v_{\mathrm{d}} v_{\mathrm{s}}} \leftarrow \mathrm{Dijkstra}(v_{\mathrm{d}}, v_{\mathrm{s}})$;
$k- = 3$;
else
choose two nodes m_1 and m_2 that are separated by $d/2+\delta$ on each direction of S, where $m_1 m m_2$ is perpendicular bisector of S;
$k- = 2$;
end
$p_{m_1 v_{\mathrm{s}_1}} = \mathrm{SourceTree}_{v_{\mathrm{s}_1} m_1} \leftarrow \mathrm{Dijkstra}(m_1, v_{\mathrm{s}_1})$;
$p_{m_2 v_{\mathrm{s}_2}} = \mathrm{SourceTree}_{v_{\mathrm{s}_2} m_2} \leftarrow \mathrm{Dijkstra}(m_2, v_{\mathrm{s}_2})$;
$p_{m_1 v_{\mathrm{d}_1}} = \mathrm{SourceTree}_{v_{\mathrm{d}_1} m_1} \leftarrow \mathrm{Dijkstra}(m_1, v_{\mathrm{d}_1})$;
$p_{m_2 v_{\mathrm{d}_2}} = \mathrm{SourceTree}_{v_{\mathrm{d}_2} m_2} \leftarrow \mathrm{Dijkstra}(m_2, v_{\mathrm{d}_2})$;
while $k > 0$ **do**
segment S = newest established path;
choose one node m_k that is separated by distance $d+\delta$ from S on the farther direction from the absolute shortest path;
$p_{m_k v_{\mathrm{s}_k}} = \mathrm{SourceTree}_{m_k v_{\mathrm{s}_k}} \leftarrow \mathrm{Dijkstra}(m_k, v_{\mathrm{s}_k})$;
$p_{m_k v_{\mathrm{d}_k}} = \mathrm{SourceTree}_{m_k v_{\mathrm{d}_k}} \leftarrow \mathrm{Dijkstra}(m_k, v_{\mathrm{d}_k})$;
$k- = 1$;
end
if *k is odd* **then**
$p_2 = p_{m_1 v_{\mathrm{s}_1}} + p_{m_1 v_{\mathrm{d}_1}}$;
$p_3 = p_{m_2 v_{\mathrm{s}_2}} + p_{m_2 v_{\mathrm{d}_2}}$;
...
$p_k = p_{m_{k-1} v_{\mathrm{s}_{k-1}}} + p_{m_{k-1} v_{\mathrm{d}_{k-1}}}$;
remove path that fails the skew requirement.;
else
$p_1 = p_{m_1 v_{\mathrm{s}_1}} + p_{m_1 v_{\mathrm{d}_1}}$;
$p_2 = p_{m_2 v_{\mathrm{s}_2}} + p_{m_2 v_{\mathrm{d}_2}}$;
...
$p_k = p_{m_k v_{\mathrm{s}_k}} + p_{m_k v_{\mathrm{d}_k}}$;
remove path that fails the skew requirement.;
end
return $(p_1, p_2, ..., p_k)$
end

Algorithm 3: Modified iWPSP heuristic for flow optimization

GeoDivRP provides the best possible path sets returned by the optimization process.

Given the capacity bound u_e on link e, we use the M/M/1 queuing model [169] that states the average packet delay on link e as

$$l_e = \frac{1}{u_e - y_e} \tag{5.5}$$

where $y_e = \sum_{w \in W} \sum_{p \in P^w} \eta_e^w(p) x_p$ is the link flow on link e, Then, the average queueing delay l_p^w for path p in commodity w is the sum of the average queueing delay on each link given by

$$l_{p^w} = \sum_{e \in E} \eta_e^w(p) l_e \tag{5.6}$$

Therefore, the average end-to-end delay for a commodity w is given by:

$$l^w = \frac{1}{q^w} \sum_{p \in P^w} \sum_{e \in E} x_p^w \eta_e^w(p) l_e \tag{5.7}$$

Based on the delay for each path for commodity w, we formulate the path skew t^w as:

$$t^w = \sum_{i \in I} |l_{p_s^w} - l_{p_i^w}| \tag{5.8}$$

where p_s^w is the shortest path for a commodity w, and p_i is the path set I that excludes p_s^w. The overall path skew T for all commodities is then given by

$$T = \sum_{w \in W} t^w \tag{5.9}$$

On the other hand, the total packet delay in the network [169] is given by

$$L = \sum_{e \in E} \frac{y_e}{u_e - y_e} \tag{5.10}$$

Based on the delay and skew, we formulate the optimization problem as follows:

$$\min \quad [(1 - \gamma)L + \gamma T] \tag{5.11}$$

subject to

$$\sum_{p \in P^w} x_p^w = q^w, \quad w \in W \tag{5.12}$$

$$\sum_{w \in W} \sum_{p \in P^w} \eta_e^w(p) x_p^w \leq u_e, \quad e \in E \tag{5.13}$$

$$x_p^w \geq q^w / k^w, \quad p \in P^w, \ \ w \in W. \tag{5.14}$$

The objective function in (5.11) targets at minimizing the delay-skew with a tuning parameter γ ($0 \leq \gamma \leq 1$), which controls the weight on either the delay or skew in the optimization process. The constraints are the same as the ones used in the minimum-cost optimization discussed earlier.

5.2.3 Complexity Analysis

We use the *ralg* solver that comes with the *OpenOpt* optimization framework [30]. The complexity for solving the flow-geodiverse linear optimization problem is polynomial. Therefore, the complexity of the GeoDivRP routing with minimum-cost optimization is dominated by the complexity of the GeoPath calculation. On the other hand, the delay-skew optimization problem is a nonlinear problem that is typically solved using an iterative process, and thus cannot be directly analyzed from a complexity point of view. We can, however, comment on the cost of solving such a problem. The total number of variables for the delay-skew optimization problem is the number of commodities plus the number of links for each topology; it is represented as nVariables $= W + E$. The current implementation of *ralg* stores a matrix of size nVariables^2 in memory, and each iteration consumes $5 \times$ nVariables^2 multiplication operations. For example, when optimizing a network with 100 commodities and 100 links, the matrix size is $200 \times 200 = 40,000$. Each iteration of the optimization has $5 \times 200^2 = 200,000$ multiplication operations. We set the max-iteration of the solver as 1000, which means the worst-case complexity

is 0.2×10^9 multiplication operations in total; this is too complex for large real-world networks. A possible improvement can be instead of waiting for the optimization to finish, the GeoPaths are returned to ResTP immediately after calculation, and the optimization is running in parallel. The necessary adjustment for traffic allocation is sent to ResTP after the optimization is done. Another possible improvement is a distributed algorithm for each commodity. We leave the detailed implementation of the two improvement mechanisms for future work.

5.3 Real-World Network Results

In this section, we present the simulation results for ResTP–GeoDivRP when using our optimization modules. We select the failure regions identified from Chapter 3. After solving the path geodiverse problem (PGD), the GeoPaths are sent to the optimization engine as input, the paths along with their flow allocation information are returned to ResTP for multiple subflow setup. These optimized paths are used for data transmission to meet the traffic demand of all the commodities. This mechanism ensures that the GeoPaths can achieve the optimal link utilization with a focus either on minimum-cost or controlled delay-skew.

In this work, we apply an area-based challenge in several physical topologies to study the link utilization and delay-skew result of GeoDivRP with the optimization engine. We also study the performance of multipath forwarding when the cascading challenge profile occurs. The steps for the routing algorithm to calculate the GeoPaths is shown as follows:

- Obtain the geodiverse paths using the modified iWPSP routing heuristic for each node pair that satisfies the skew constraint and d-distance separation criteria.

- Solve the multi-commodity flow optimization using the linear programming formulation (LP) or nonlinear programming formulation (NLP) for the flow-diverse minimum-cost or delay-skew optimization, respectively.

- Use the optimized GeoPaths in *ns-3* network simulations.

5.3.1 Optimization Results

We now present the flow optimization result for both of the minimum-cost and delay-skew case. The topologies considered are the structural physical graphs [24] with their properties shown in Table 5.2. We include CORONET [160], Internet2 [158], Level 3 [151], Sprint [24], and TeliaSonera [158] fiber-level networks. CORONET is a synthetic fiber network to represent Internet service provider topology. The number of nodes and links for the considered topologies are in the same range, and the average node degree for all the topologies are about three. The capacity for all the links is set to 5 Gb/s, and we use constant bit rate (CBR) traffic, sent from each node to all the others at a data rate of 10 Mb/s as the traffic demand. We use the challenged area at Kansas City identified in our critical-region identification mechanism presented in Chapter 3 with a 300 km radius.

Table 5.2: Physical topology analysis

Network	**Nodes**	**Links**	**Degree**	**Diameter**	**Radius**	**Path Length**
CORONET	39	63	3.23	9	5	4.08
Internet2	16	24	3.00	6	3	2.63
Level 3	63	94	2.98	14	7	5.68
Sprint	77	114	2.96	16	9	6.47
TeliaSonera	18	21	2.33	7	5	3.58

Minimum-cost optimization

We record the solving time of the optimization problem for different topologies. As shown in Table 5.3, the maximum time for the optimization is about 7 s for the Sprint network, while most of the others take less than 1 s. The evaluation is carried on a Linux machine with a 3.16 GHz Core 2 Duo CPU and 4 GB memory.

Table 5.3: Execution time for optimization algorithm

Network	**Number of Nodes**	**Number of Links**	**Number of Failed Nodes**	**Optimization Time (s)**
CORONET	39	63	2	0.62
Internet2	16	24	1	0.04
Level 3	63	94	4	2.06
Sprint	77	114	3	6.96
TeliaSonera	18	21	1	0.02

We further compare GeoDivRP to OSPF in terms of the overall link-congestion factor. Recall that the link-congestion factor is defined as the percentage of the bandwidth that has been used by the network flows. Our minimum-cost optimization formulation is not specifically minimizing the link-congestion factor; therefore, some links are still using up to 100% link capacity. However, since we specify the capacity upper bound on path flows, GeoDivRP uses the network resources efficiently and does not congest any network link. For OSPF, on the other hand, the model always selects the shortest path without considering the remaining network resources, which causes congestion by overloading some network links. In the network simulation context, the extra traffic assigned to network links will either be dropped or queued if router buffers are used; traffic loss or increased delay will occur, respectively.

In Figure 5.3, we present the link-congestion factor for the Level 3 network when the demand is 10 Mb/s for each node pair. GeoDivRP does not overload any link by

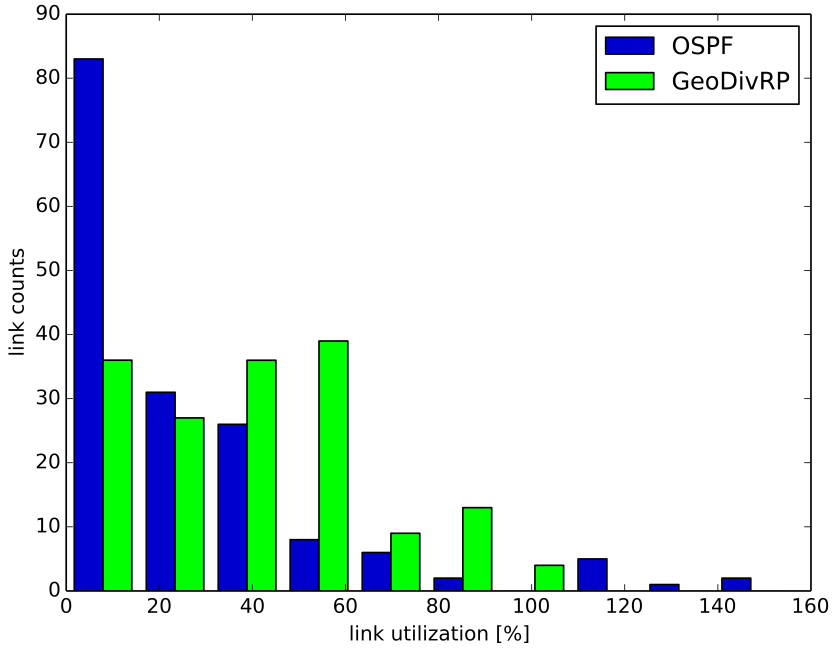

Figure 5.3: Link utilization in Level 3 network

distributing the traffic load among multiple paths. However, OSPF has utilized some links up to 140%, which means that for each of the overloaded link, 40% of the data traffic will be dropped. As the link capacity is 5 Gb/s, 2 Gb of traffic is dropped each second on these overloaded links. This causes significant traffic loss to the network communication, and it is especially damaging when the network is under large-scale challenges. The dropped traffic could have been buffered but the end-to-end delay would increase exponentially. Our delay-skew optimization targets at minimizing the path delay and skew and we present its result in the next subsection. The network-congestion factor for GeoDivRP is 100% while that for OSPF is 140%.

This link-congestion analysis has demonstrated that GeoDivRP with flow-geodiverse minimum-cost optimization can allocate traffic to multiple paths efficiently and avoid overloading any network link; while OSPF over-utilizes links and causes the data packets to either be dropped or buffered with increased end-to-end delay.

Delay-skew optimization

For the delay-skew optimization scenario, we set the link capacity at 500 Mb/s, with demand at 10 Mb/s. The total number of commodities is 9. We choose the source nodes in the west coast sending to destination nodes in the east coast. This way the paths calculated represent the highest delay scenario. The GeoPaths provided by iWPSP are the paths calculated based on the current network topology with area-based challenges. The challenge region is the same as that for the minimum-cost optimization at Kansas City. For each commodity, we calculate three geodiverse paths for optimization.

Similarly, we record the time for solving the delay-skew optimization problem in different physical topologies, as shown in Table 5.4. The evaluation is performed on a Linux machine with a 3.16 GHz Core 2 Duo CPU and 4 GB memory, same as the minimum-cost optimization case. All the physical topologies have a reasonable optimization time for both the single pair and nine node pairs cases. For the nine-node-pairs case, it takes five seconds to solve the problem for Sprint, which is the maximum time among all the topologies as it is the largest one considered. The time for a single traffic pair is below 1 s for all the topologies. This means that a distributed algorithm for the delay-skew optimization is durable for the real-world network communication. It is planned for future work.

Table 5.4: Time for delay-skew optimization algorithm

Network	**# of Nodes**	**# of Links**	**# of Failed Nodes**	**# of Commodity**	**Single Pair Time (s)**	**Nine Pair Time (s)**
CORONET	39	63	2	9	0.87	4.13
Internet2	16	24	1	9	0.51	3.62
Level 3	63	94	4	9	0.53	8.30
Sprint	77	114	3	9	0.81	5.04
TeliaSonera	18	21	1	9	0.52	3.63

We further carry out simulations with a varying traffic demand and study the largest demand that GeoDivRP supports in a given physical topology. We present the variation of delay and skew when the demand increases for the five topologies. We do not include the delay and skew result for OSPF; the network becomes congested with low demand and the delay becomes too large to present in the same plot with GeoDivRP; γ is set as zero for the delay optimization. As shown in Figure 5.4, the demand curves for all the topologies begin with a low value around 15 ms and increases slowly when the demand increases. However, when the demand increases beyond the *demand collapse point*, the delay starts increasing exponentially until the optimization cannot provide solutions. For example, if we consider the delay curve for the CORONET network, when the demand increases from 180 Mb/s to 190 Mb/s, the delay increases from 35 ms to over 200 ms, and the network becomes too congested to provide normal service beyond the demand collapse point, which is 190 Mb/s in this case. With the different demand collapse points for the topologies provided to ResTP, better flow allocation decisions can be made and the application can use network resources more efficiently. However, the implementation of ResTP is beyond the scope of this work.

In Figure 5.5, we present the skew minimization result using the same set of topologies; γ is set as one to focus on the skew optimization. For the demand below 100 Mb/s, the skew decreases as the traffic load increases; each link has a low delay and the number of hops for each path in one commodity contributes more to the end-to-end delay. However, when the demand increases beyond 100 Mb/s, the link delay for the topologies except CORONET begins increasing exponentially. Therefore, the path skew increases exponentially as well.

We continue our simulation with the link-congestion analysis in the five topologies. The topologies and the challenge scenario are the same ones used for the previous experiment. The link capacity is set as 500 Mb/s, and the demand is 50 Mb/s; the number

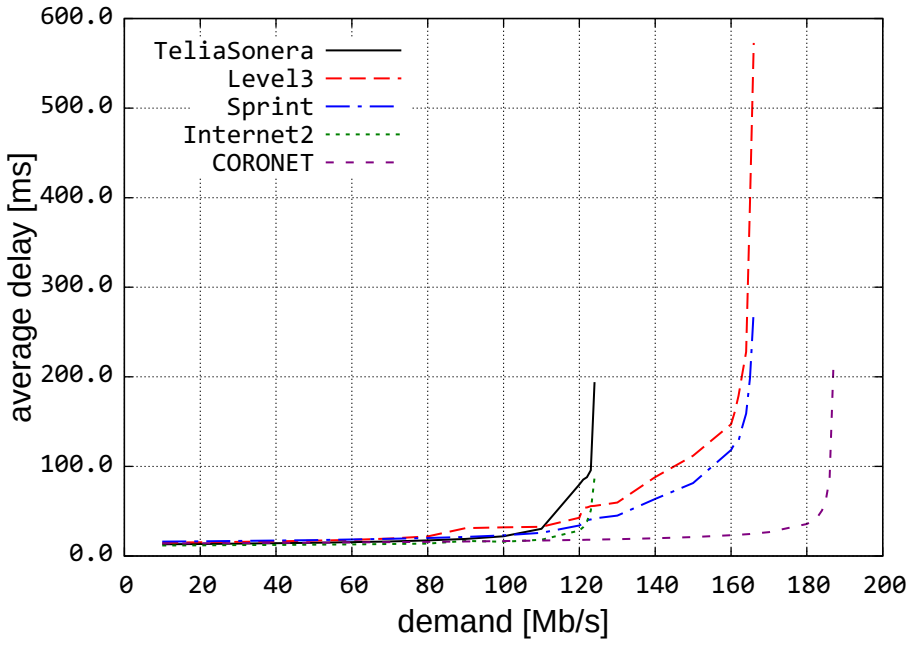

Figure 5.4: Path delay with varying demand

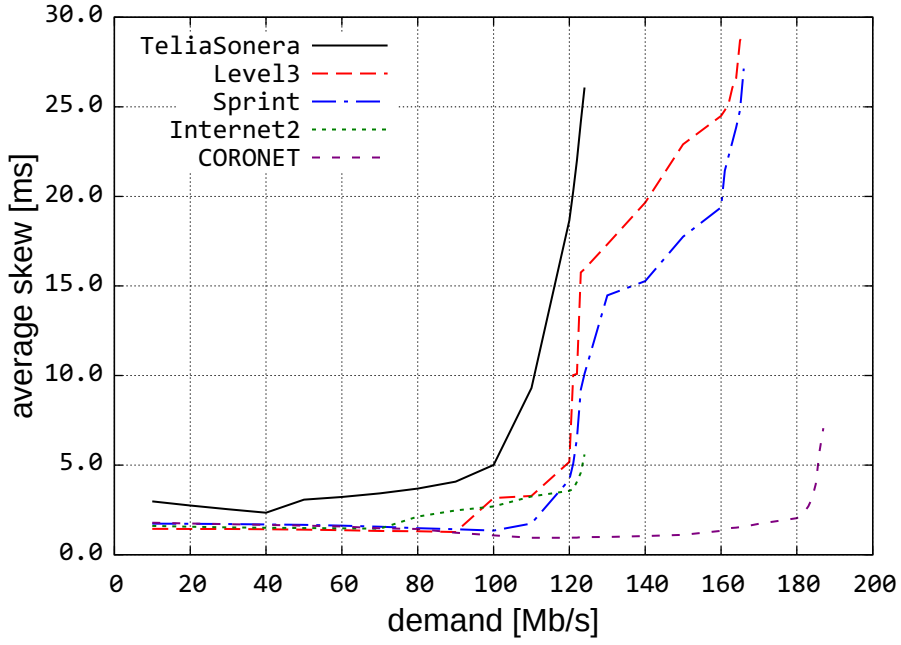

Figure 5.5: Path skew with varying demand

of commodities is 100. The reason for the demand and number of commodity choice is to have a reasonable amount of traffic going through the network to better demonstrate the effectiveness of GeoDivRP.

As shown in Figure 5.6, the x-axis presents the link utilization in percentage, and the y-axis shows the number of links with that utilization level. Take 100% link utilization as an example, OSPF has five links, while the number for GeoDivRP is six. We have added a temporary queue to each node to keep count of the packets that are overloading any links adjacent to the node. We can obtain the number of overloaded links through the queue.

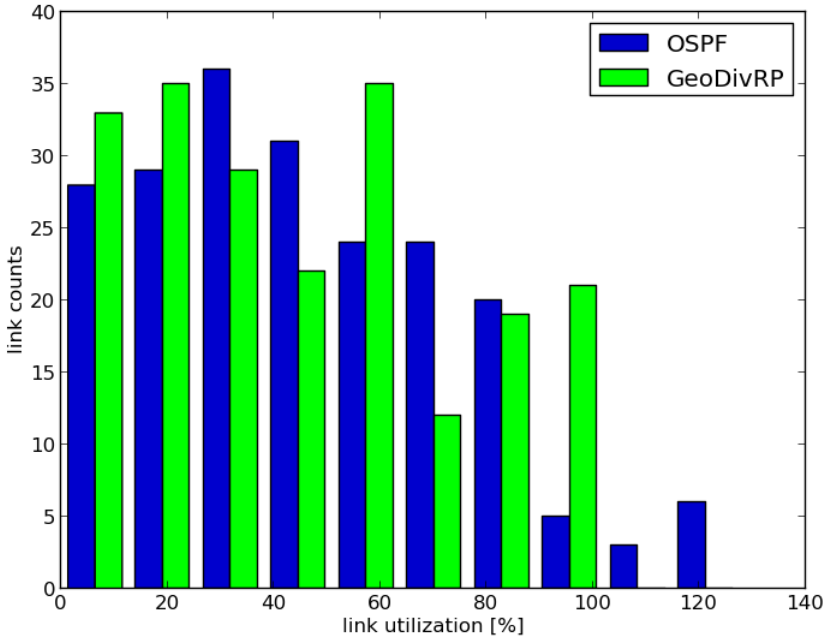

Figure 5.6: Sprint network link utilization

GeoDivRP guarantees that the link utilization for any link is not over 100% and keeps lower link usage whenever possible, which is specified by the objective function in the formulation. On the other hand, OSPF simply distributes network traffic among the calculated paths and can easily congest the network when the demand becomes larger. As

shown in Figure 5.6, OSPF congests 6 links; although this may not be a large percentage out of the 114 total links, they cause 85% of the commodities and 59% of the paths congested. On the other hand, GeoDivRP guarantees the optimized traffic allocation on all the commodities and presents great performance improvement.

As shown in Figure 5.7, GeoDivRP guarantees the link utilization is not over 100%, yet the usage for OSPF goes to 160% and therefore greatly congests the network; there are 15 congested links out of 94. Similarly, although not a large percentage, these links cause 91% of commodities and over 71% of the paths congested. On the other hand, GeoDivRP avoids congestion by optimizing the traffic allocation on multiple paths of each commodity.

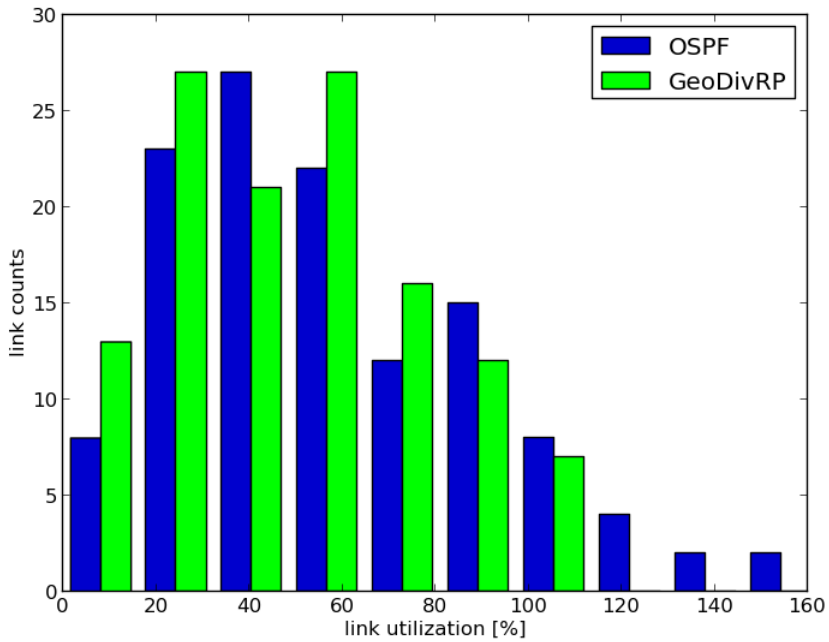

Figure 5.7: Level 3 network link utilization

The other three topologies present similar results and are shown in Appendix A.

The objective function for the delay-skew formulation balances the delay and skew

using the tuning parameter γ. In Figure 5.8, we present the average delay and skew change for a single path with the varying γ value using the CORONET network. The results for the other networks present a similar trend and are not shown. The points on the plot are the γ values ranging from 0 to 1 with 0.1 step increment. The traffic demand and link capacity are 50 Mb/s and 500 Mb/s respectively. As we observe from the figure, when γ increases, the average delay for each commodity increases while the average skew decreases. This means that delay and skew work against each other in this optimization formulation. Based on different application scenarios, our model could select different γ for better network communications. The selection of delay-skew combination is part of the ResTP design and beyond the scope of this work.

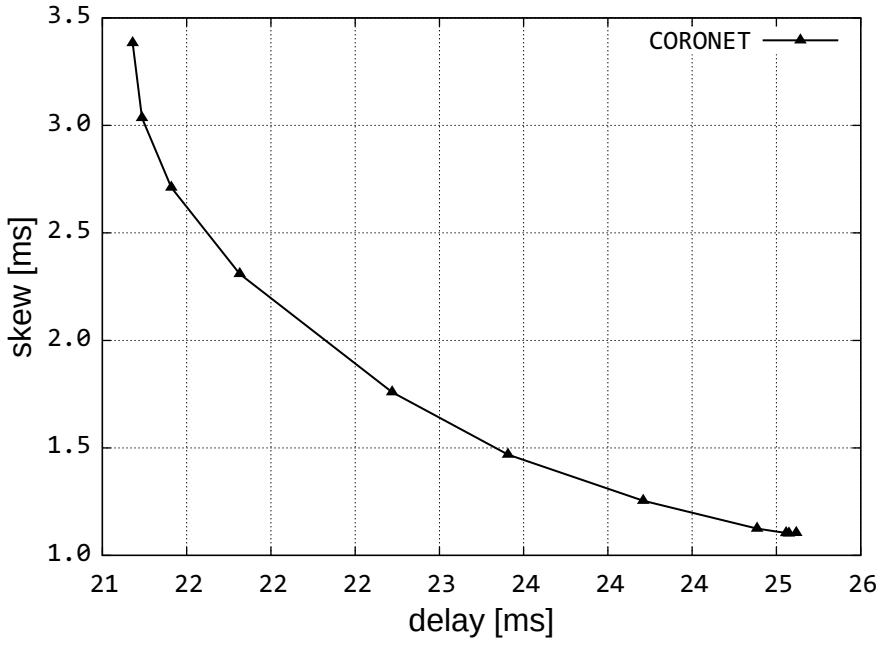

Figure 5.8: Delay and skew with varying γ for CORONET

5.3.2 Multipath Performance Comparison

We further present a comprehensive performance analysis of our ResTP–GeoDivRP protocol stack against MPTCP [26, 170] using multiple node-disjoint paths. The *ns-3* sim-

ulator is again used to demonstrate our protocol stack's performance compared to an implementation of MPTCP [170]. All the nodes in the topology are ResTP–GeoDivRP enabled, and path protection using multiple geodiverse paths is provided by GeoDivRP. Three GeoPaths are used in all the considered topologies for survivable routing. ResTP is still under active development; we include only the cross-layer path function with no congestion control mechanism.

Challenge profiles

A set of challenge profiles is used to systematically study their impact on the network connectivity and how our protocol stack performs in the face of these challenges.

- The Midwest challenge profile shown in green circles represents a super-tornado sweeping trajectory
- The coastline challenge profile along the East Coast shown in blue circles represents a hurricane trajectory
- The cascading challenge profile such as power blackout affects a region growing in size as shown in red circles

As shown in Figure 5.9, the Sprint physical network [31] is presented with several challenge profiles. The movement for the Midwest profile is from the southwest to the northeast direction representing a super-tornado, while the coastline profile moves in a similar direction but on the east coast representing a hurricane. It also has a larger challenge radius compared to the Midwest profile. The profiles provide a better understanding of how different challenge locations and trajectories affect the protocol stack performance. For example, the cascading challenge profile shown in Figure 5.9 as red circles concentrates on the infrastructure with most shortest path occurrences. It renders

the network more difficult to maintain normal network connectivity since the affected nodes are forming a large percentage of shortest paths connecting the west and the east coast. In this work, we apply a cascading challenge profile in the Sprint [31] and Level 3 networks [151], and also study how the Midwest profile affects our KanREN [29] testbed in Chapter 6.

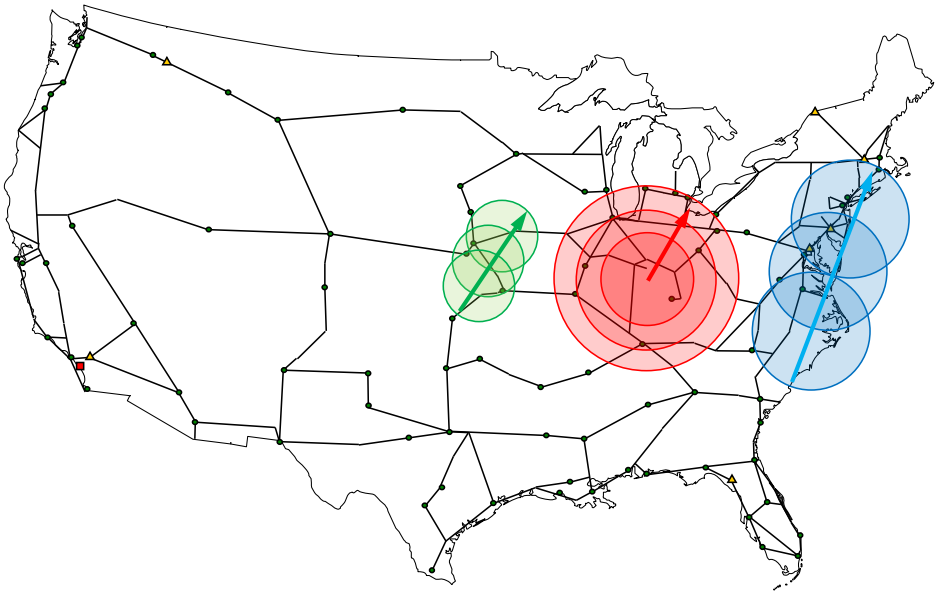

Figure 5.9: Sprint network topology challenge profile

The cascading challenge profile is applied in the Sprint network shown in Figure 5.10. In this profile, three challenge scenarios are included. The challenge begins at 20 s and grows larger in radius with each challenge lasting for 20 s. It originates around Nashville and grows larger in range; the challenge in green circle occurs at 20 – 40 s, the yellow circle challenges at 60 – 80 s, and the red one at 100 – 120 s. The traffic originates from Oklahoma City to Washington D.C. and the bandwidth on each link is 100 Mb/s. The dashed line represents the paths calculated for MPTCP. These are node-disjoint paths calculated using Suurballe's algorithm [59, 60], and it cannot guarantee all the paths are geographically disjoint. In this experiment, MPTCP uses St. Louis, Kansas City, and Atlanta as its next-hops for the three paths. The second challenge shown in yellow circle

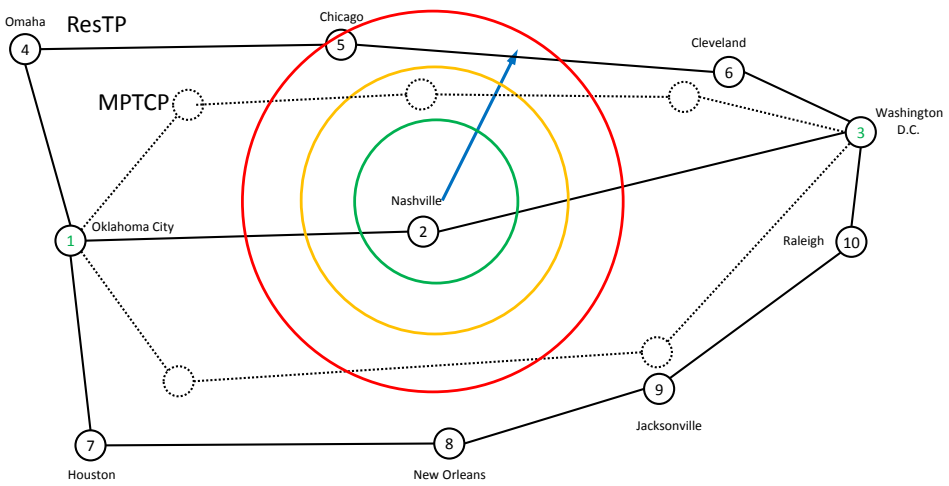

Figure 5.10: Sprint network topology under cascading challenge

fails two of MTCP's paths and the third challenge fails all three of them.

On the other hand, GeoDivRP guarantees the paths are geographically disjoint, and therefore with all the subflows created by ResTP geodiverse, the protocol stack can provide higher throughput and better resilience to cascading challenges. As shown in Figure 5.10, GeoDivRP uses Omaha, Nashville, and Houston as its next-hops. This guarantees that for any regional challenges with a radius no larger than the distance-separation criteria d, at least 1/3 paths will survive the regional challenge, and most of the time 2/3 of the paths. The traffic still originates from Oklahoma City to D.C. with the solid line representing the paths calculated for ResTP. The paths are provided by GeoDivRP using iWPSP heuristic [103] with the d-distance separation guaranteed.

Figure 5.11 plots the average throughput in terms of Mb/s across the three paths against the simulation time. The throughput starts from zero and approaches 70 Mb/s at the beginning of the simulation until the first challenge occurring at 20 s. MPTCP does not guarantee the distance separation among the multiple paths used in the simulation; therefore, with circular radius d challenge, each circular failure can take down two or all paths at the same time if occurs at the right location. From 20 – 40 s, with the shortest

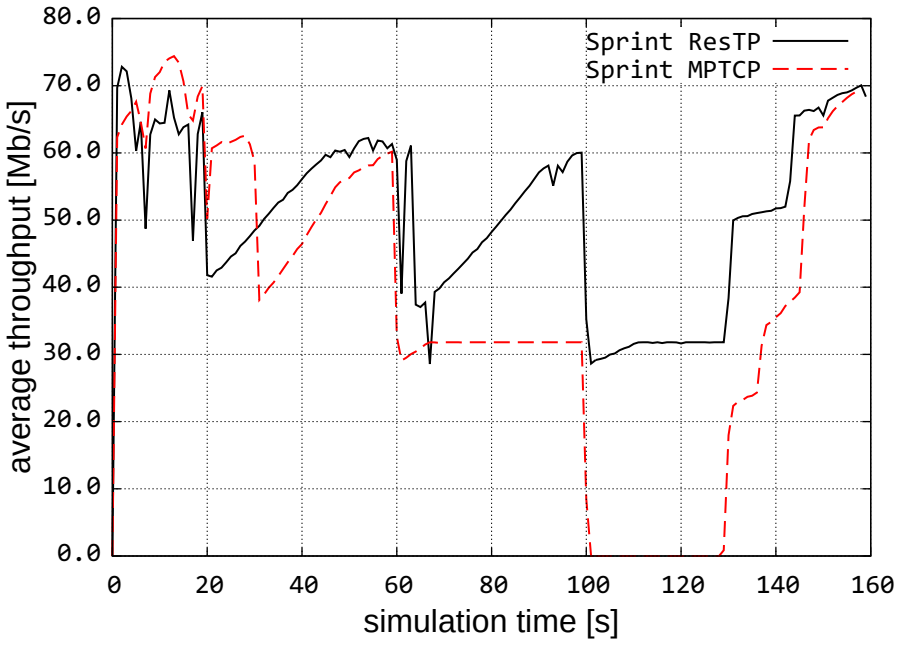

Figure 5.11: Sprint network ResTP throughput compared to MPTCP

path failed, both ResTP and MPTCP reduce in throughput. From 60 – 80 s, ResTP obtains 2/3 of the full average throughput while 1/3 for MPTCP since two of its paths are failed. The worst performance for MTCP occurs from 100 – 120 s as all three of its paths are failed. Overall, ResTP presents around 30% to 40% performance increase compared to MPTCP in face of regional challenges.

A similar challenge profile is applied in the Level 3 network [151] as shown in Figure 5.12. The failure region is shown in the KU TopView page [31, 54] to better present the overall topology and how the challenge affects nodes and links. Similar to the previous experiment, the failures in color-coded circles represent the cascading challenge growing in size. Two red dots represent the source node at Denver, CO and destination node at Indianapolis, IN. The outbound red arrow from Denver shows the shortest path, while the two green arrows represent the two alternative paths calculated by GeoDivRP.

As shown in Figure 5.13, when the first challenge is introduced at 20 s, both ResTP and

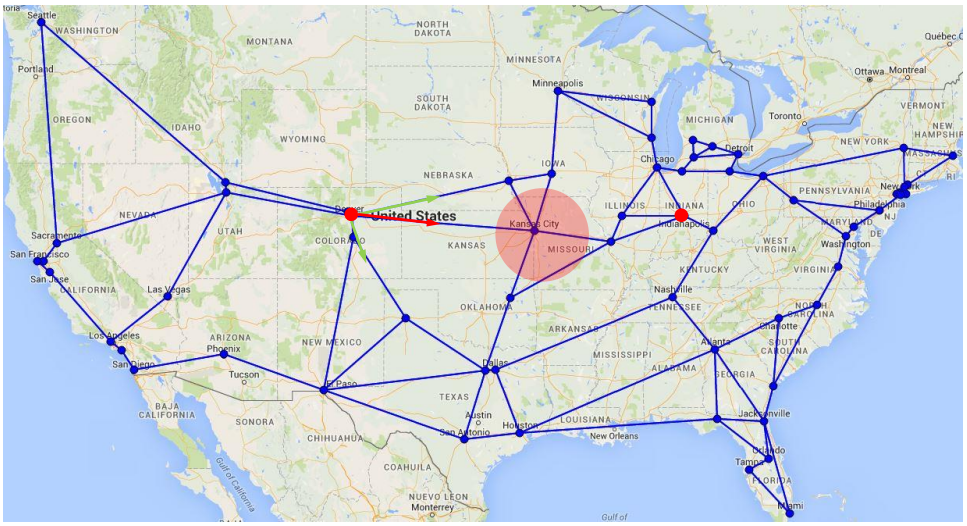

Figure 5.12: Level 3 cascading challenge scenario

MPTCP see a drop in throughput. At 60 s, ResTP is able to restore 2/3 the throughput by using the geodiverse path provided by GeoDivRP. On the other hand, the throughput for MPTCP further reduces as it lost another path. The worst performance for MPTCP occurs at 100 – 120 s since all its three paths are failed during this period, while ResTP with cross-layer path information can still achieve 30 Mb/s throughput. After 120 s with all the challenges elapsed, both of the protocols restore back to normal operation.

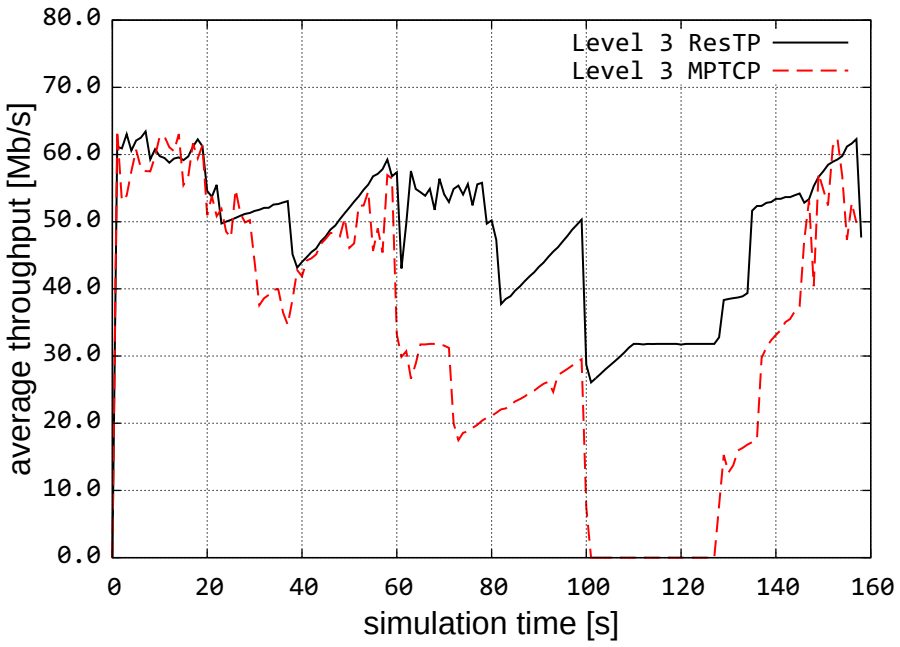

Figure 5.13: Level 3 network ResTP throughput compared to MPTCP

Chapter 6

SDN Resilience Experiments

In this chapter, we extend our work to the software-defined networking (SDN) domain to further analyze the performance of our ResTP–GeoDivRP protocol stack in Section 6.1. We extend an OpenFlow controller [145] to achieve the GeoDivRP routing mechanism by utilizing the geodiversity in the network topology. We further analyze the performance of ResTP–GeoDivRP in the real-world network topologies in Section 6.2 and Section 6.3 for our *Mininet* and testbed experiments, respectively.

6.1 Web Framework Design

We implement our Web framework in SDN with a fullstack design. The frontend represents network topologies on Google Map and a user-tunable polygon represents a failure region. The backend OpenFlow module emulates network challenges using Mininet or running experiments on our KanREN OpenFlow testbed. We have presented a demo for the real-time operation of our framework at the 23rd and 24th GENI Engineering Conference [171]. As shown in Figure 6.1, the frontend system reads the adjacency matrix from KU-TopView [31, 54] and creates the topology automatically by overlaying it on top of the Google Map. If the backend system is powered by Mininet, the topology is

automatically created with realistic delay and bandwidth configurations as an Mininet experiment. OpenFlow switches are used to represent network nodes using the nodes' actual physical coordinates in the topology. The users interact with the system through a drag-and-drop polygon representing the challenge region. The polygon can be modified to any shape or size by the users, which causes the links and nodes to fail if covered by the polygon (challenge). The challenge information is then passed to the backend system which runs the OpenFlow experiments either on our testbed or Mininet. Physical OpenFlow switches are deployed in our OpenFlow testbed, while Mininet-emulated [28] topologies are running on our backend system for all the other network topologies. We have implemented our frontend and backend system and present our website at [OpenFlow Demo].

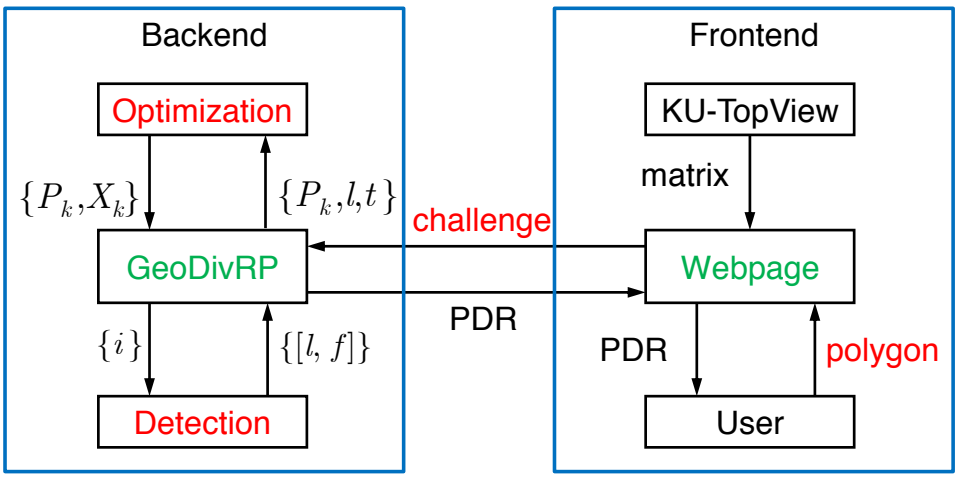

Figure 6.1: Web framework for challenge emulation

In our testbed, we have deployed OpenFlow-enabled switches in Kansas Research and Education Network (KanREN) [www.kanren.net], which is a logical ring throughout the state of Kansas connecting institutions of higher education. Eight Brocade NetIron CES 2024C [172] OpenFlow switches have been deployed at these institutions, as shown in Figure 6.2. A full-mesh topology is deployed as an OpenFlow overlay and any arbitrary virtual topologies can be initialized through Multiprotocol Label Switching (MPLS) [173]

tunnels. A ring topology is used in our experiment. Floodlight [174] is an OpenFlow controller based on Java, and it works with both physical- and virtual-switches. Our resilience routing framework controls the switches through Floodlight.

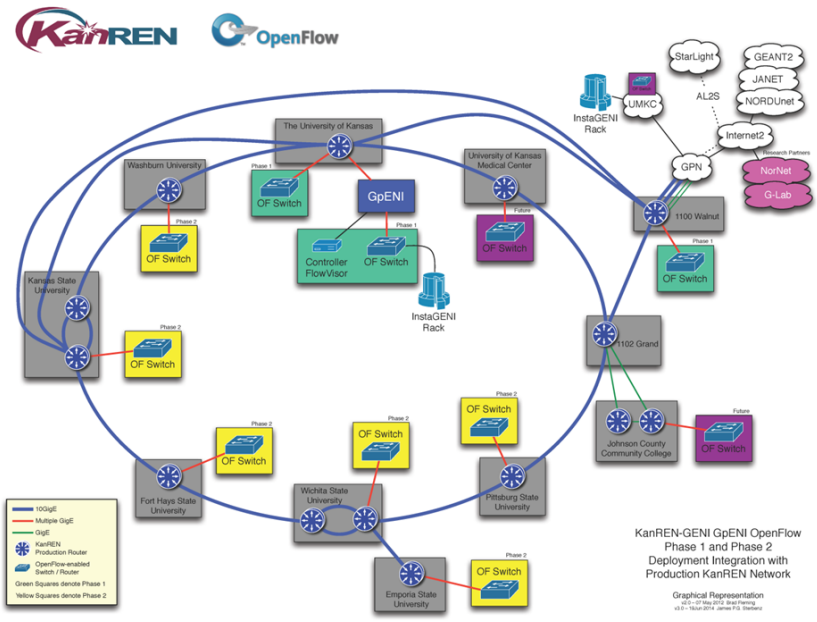

Figure 6.2: KanREN OpenFlow switches deployment

The Mininet emulator is running in our backend system to emulate large-scale network topologies that beyond the capacity of our testbed. To experiment Sprint network topology for example, after the user select it from our frontend system, the specified topology will be deployed automatically in our backend servers using Mininet and corresponding performance metrics are provided in our frontend system, such as the throughput for TCP or packet delivery ration (PDR) for UDP. Along with our physical testbed, it consists a real-time challenge-analysis system to provide quick performance analysis for researchers or students in class.

6.1.1 Backend Model Implementation

Our protocol stack acts as the backend system and powers the resilient experiments. GeoDivRP is implemented as an extension to the OpenFlow controller; with the centralized view of the topology from the SDN environment, the failed node and link information is used to notify GeoDivRP about the current network condition. Based on the real-time information, GeoDivRP is able to provide multiple GeoPaths for dependable communication.

The major change to our protocol stack in SDN is the monitor module. As shown in Figure 6.3, link failure monitor module collects network statistics and provides the link information l and the failure information f to GeoDivRP. GeoDivRP acts on this information and makes routing decisions such as which GeoPath to choose. Network statistics are acquired using the OpenFlow discovery protocol (OFDP). The network devices advertise their link capacity and the controller constructs a centralized layer-2 network topology.

6.2 Mininet Experiment

In this section, we present the emulation results using Mininet. The experiment begins with reading the adjacency matrix for different physical topologies and creating Mininet experiments programmatically with realistic delay and bandwidth configurations. The bandwidth used for this experiment is a uniform 100 Mb/s[1] across all links and realistic delay parameters are chosen based on the physical distance between the respective hosts. OpenFlow switches are used to represent network nodes in the physical topologies.

The Sprint physical topology is used in this experiment with nodes shown in blue

[1]Bandwidth can be realistic values if provided

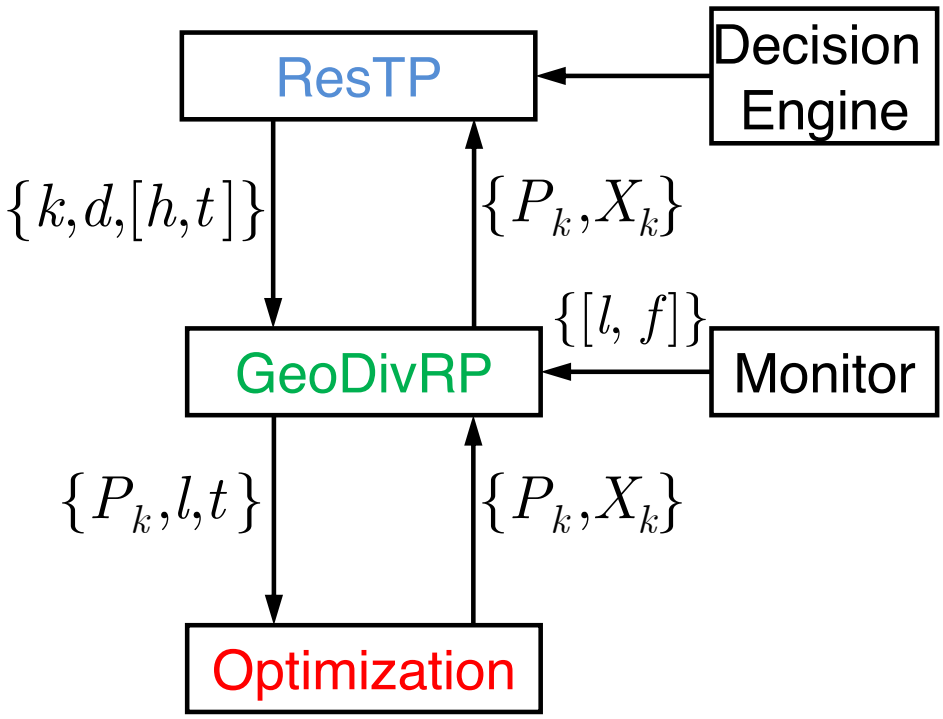

Figure 6.3: GeoDivRP and optimization engine

dots and links in green straight lines in Figure 6.4 and 6.5. The red polygon represents the challenge region tunable by the users. Internet Control Message Protocol (ICMP) messages are used to evaluate the performance, with its packet delivery ratio (PDR) displayed on our website in real-time when the experiment is running. The traffic is sent from Seattle, WA to New York City, NY and Los Angeles, CA to Miami, FL for each of our scenarios. When the regional challenge occurs at Chicago and later at Dallas, the traffic is rerouted around the challenge and a new path is calculated by the controller. The end-to-end delay for the above experiment is shown in Figure 6.6. The initial delay spike is caused by path discovery in both of the cases. The delay for both scenarios is in the range of 50–60 ms for the next 24 packets when the network is unchallenged. The challenge is applied at the 26th packet and the new path discovery causes the delay spike shown in the middle of the graph. Once the challenge is applied, due to rerouting, the delay for the next set of packets is higher than the unchallenged ones.

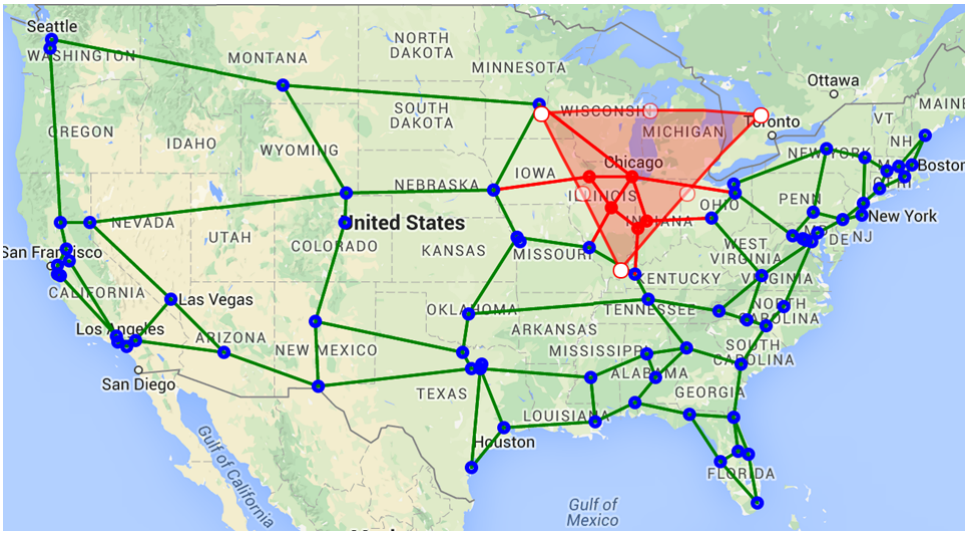

Figure 6.4: Sprint network failure scenario one

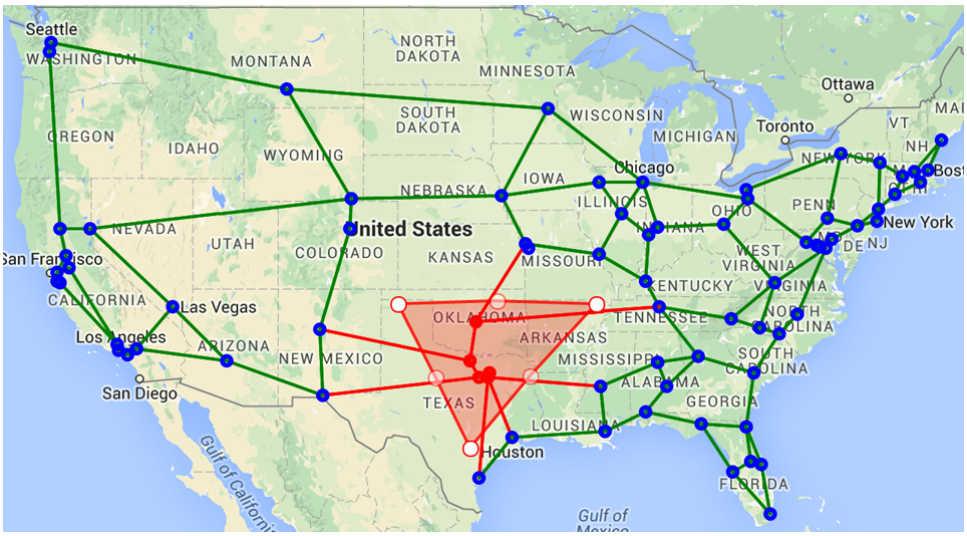

Figure 6.5: Sprint network failure scenario two

6.2.1 MPTCP Experiments

We then study the switch topologies under challenges with Multipath TCP (MPTCP) [25]-enabled routers and a single sender and receiver. We use a kernel version of MPTCP [26]. All the links' bandwidth are 10 Mb/s. The topology for the experiment is presented in

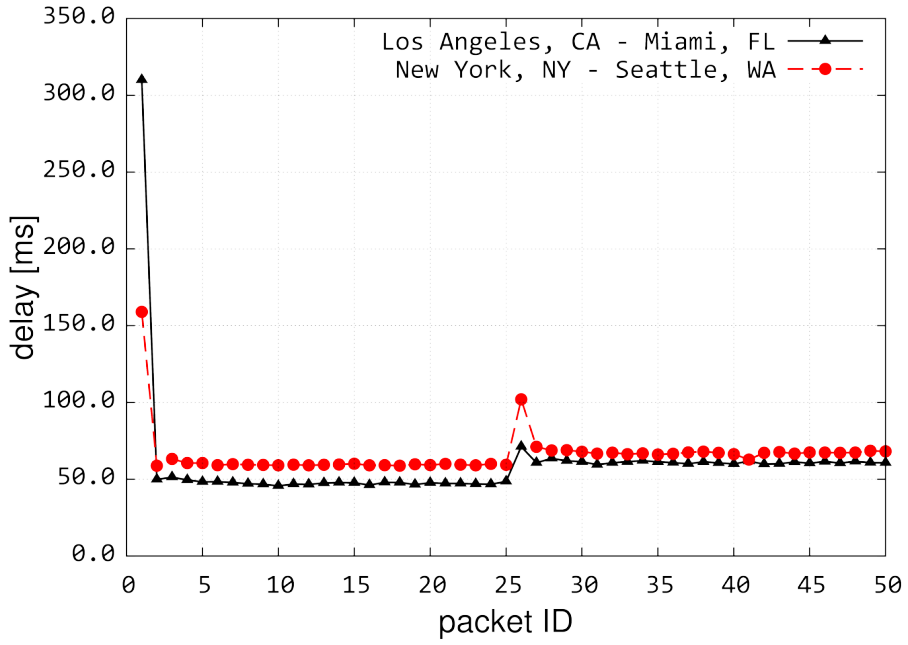

Figure 6.6: Sprint OpenFlow switches delay

Figure 6.7 where multiple paths exist between Lawrence and Wichita. A challenge profile in the Midwest is applied over the topology starting from Pittsburg and moving towards Topeka. The initial challenge takes effect at 30 s bringing down the Pittsburg switch. The next challenge occurs at Emporia starting at 60 s with the Pittsburg switch brought up. For the final challenge, the challenge circle encompasses Topeka at 90 s with the Emporia switch brought up again. Finally, the challenge circle moves away from the topology with all switches up at that time.

Results from the above challenge profile is shown in Figure 6.8. The traffic is generated using iPerf [175], a network framework for evaluating the network's maximum bandwidth. For the first 30 s, the throughput for different cities are close to 10 Mb/s, the link capacity. Starting at 30 s, the throughput of Pittsburg drops as the challenge is over Pittsburg. At the end of 60 s, Emporia drops off the network while Pittsburg is brought up, which explains the rise in throughput for Pittsburg. After another 30 s, the challenge moves

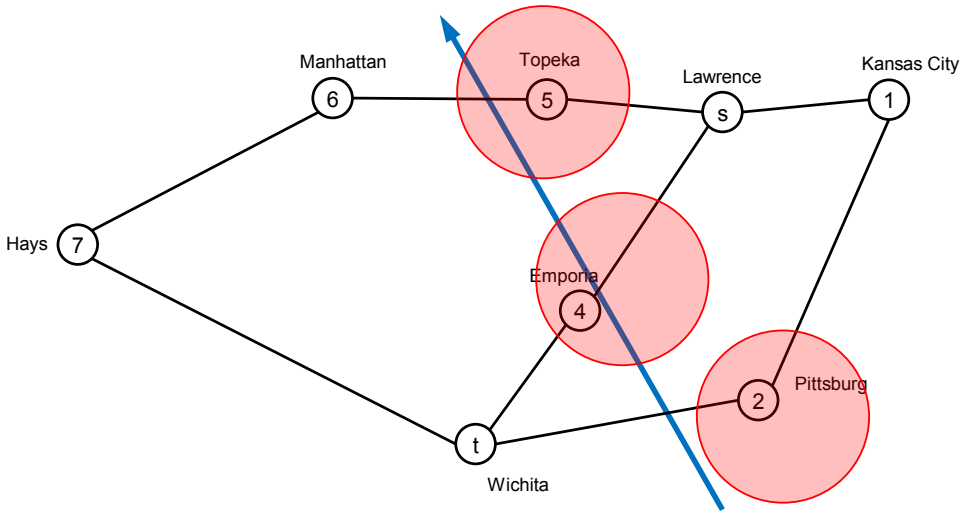

Figure 6.7: KanREN OpenFlow network regional challenges

away from Emporia shown by the rise in throughput at 90 s and Topeka is challenged. After 30 s, the challenge moves away from Topeka shown by the rise in throughput at 120 s.

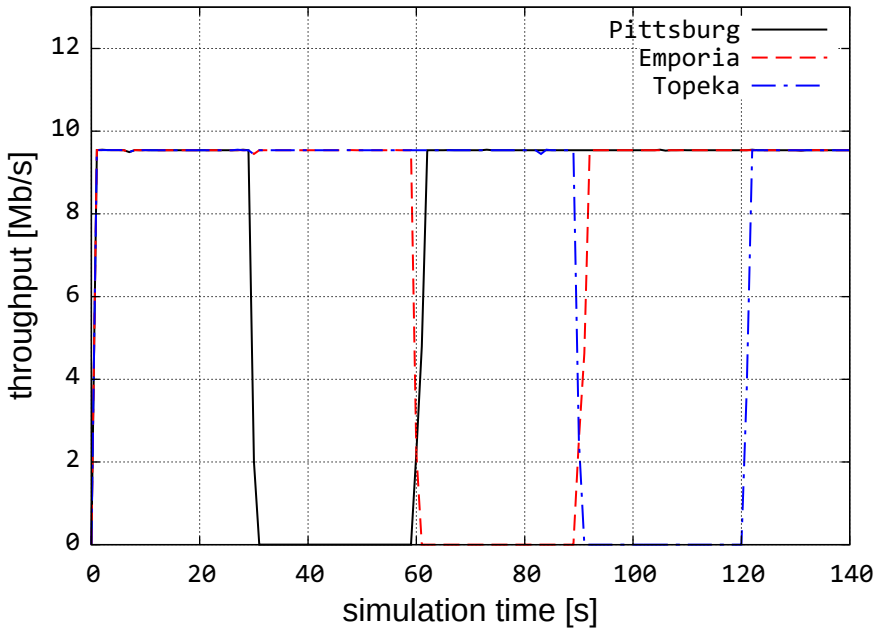

Figure 6.8: KanREN MPTCP throughput result

6.3 KanREN Testbed Experiments

The KanREN testbed experiments represent how our protocol stack works in real-world OpenFlow switches. Through testbed experiments, we can study geo-correlated challenge characteristics and analyze how our protocol stack performs. As shown in Figure 6.9 and 6.10, the blue dots represent the Brocade OpenFlow switches and green solid lines the links. The traffic is sent from Lawrence, KS to Kansas City, KS in our first scenario and Wichita, KS to Pittsburg, KS in the second. When the challenge takes down Wichita–Pittsburg and Lawrence–KC link for each of our aforementioned scenarios, the traffic reroutes around the failure regions and the average rerouting delay is presented in Figure 6.11. The trend for the end-to-end delay is similar when comparing both challenge scenarios. We observe an early high delay for the initial sample which is due to the initial packet trying to find the path to the destination. The next 24 packets have an average delay of 1 ms for Lawrence–KC and 4 ms for Wichita–Pittsburg. The challenge is applied at the 26th packet and is clearly shown by the middle delay spike in both of the challenge scenarios. Rerouting by the controller occurs and packets are routed through an alternate path with higher hops and higher delay than the unchallenged case in both of our scenarios.

6.3.1 ResTP-GeoDivRP Results

We study our protocol stack's performance at our KanREN physical testbed using two failure scenarios as presented in Figure 6.12 and Figure 6.13. The traffic originates from Lawrence to Wichita with all the links' bandwidth at 100 Mb/s and there are multiple paths exist. A Midwest challenge profile is applied over the topology starting from Pittsburg and moving towards Topeka shown as the blue arrow.

For the smaller failure case as shown in Figure 6.12, the initial failure f_1 takes effect

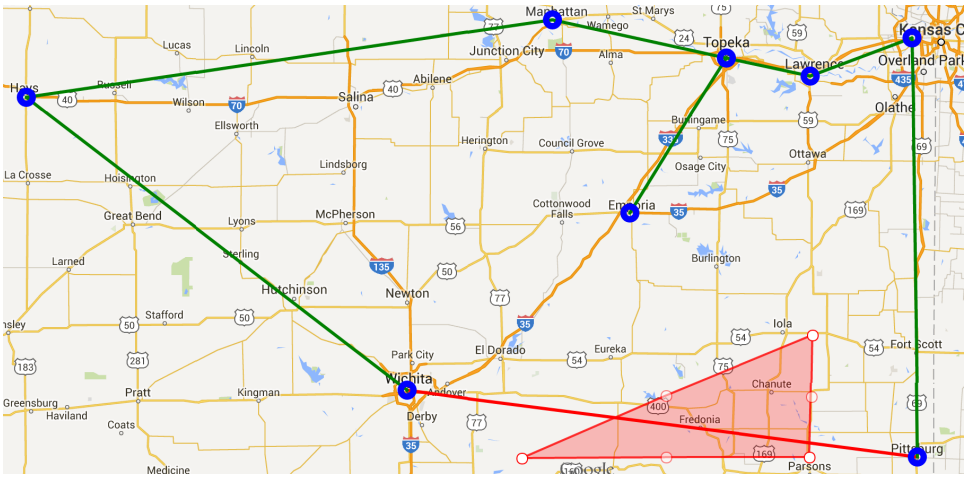

Figure 6.9: KanREN OpenFlow testbed failure scenario one

Figure 6.10: KanREN OpenFlow testbed failure scenario two

at 20 s bringing down the Pittsburg switch and lasts for 20 s. The next failures f_2 and f_3 occur at the Emporia switch starting at 60 s lasts for 20 s as well. The last failure circle f_4 encompasses Topeka at 100 s for 20 s. Finally, the failure moves away from the topology with all switches up after 140 s. Throughout the challenge, only one path is failed at any given time. We are using multiple TCP connections to emulate ResTP abilities in our testbed.

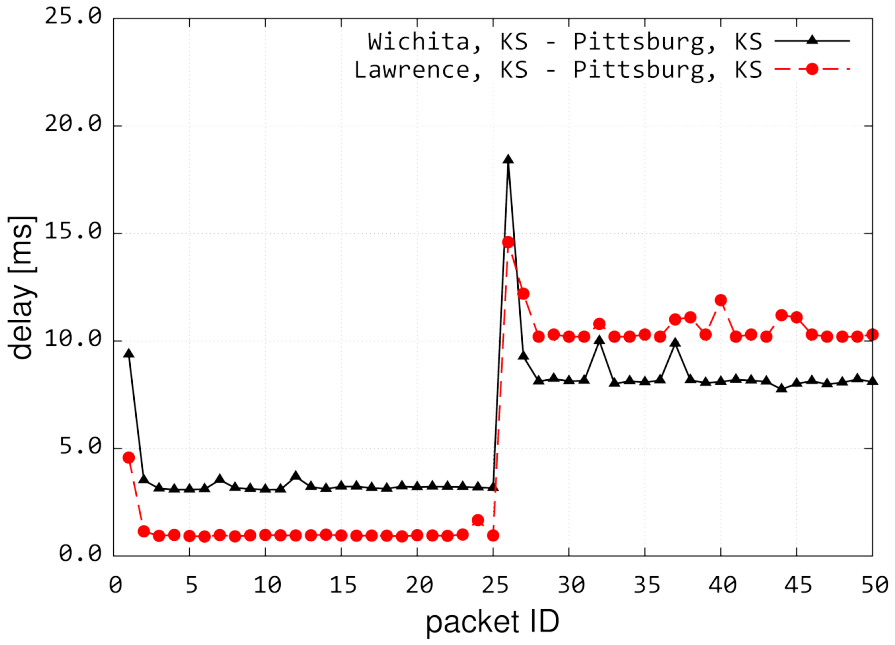

Figure 6.11: KanREN OpenFlow switches delay

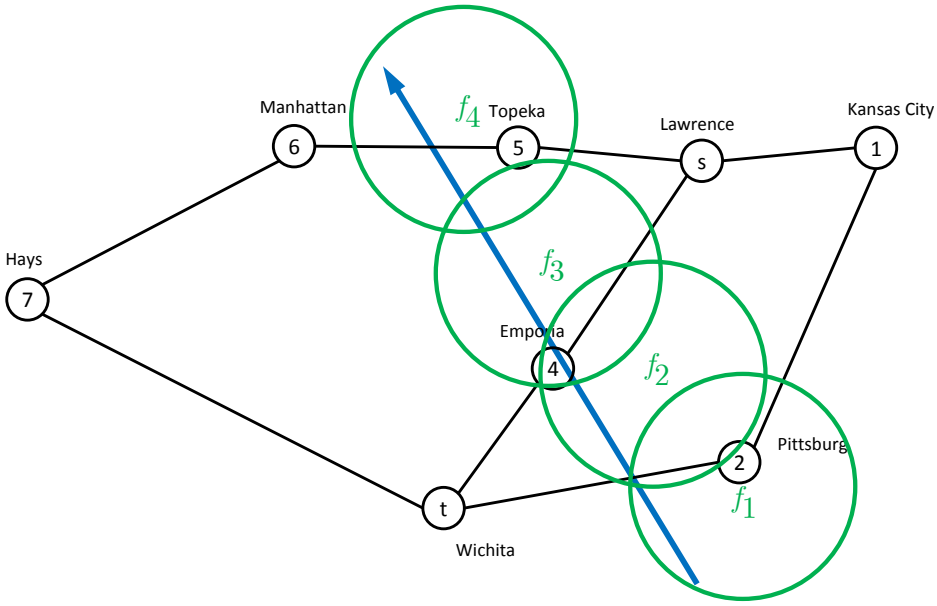

Figure 6.12: KanREN small failure radius

For the larger failure radius shown in Figure 6.13. The challenge begins at Pittsburg and follows the same trajectory as the previous case. Each challenge lasts for 20 s as

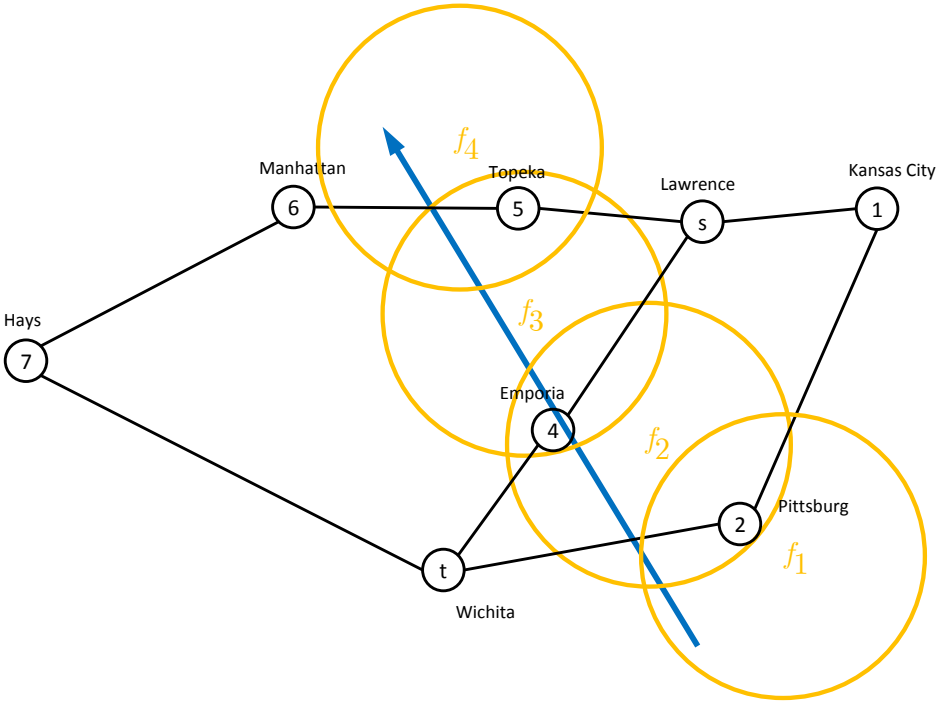

Figure 6.13: KanREN large failure radius

well. But with the larger radius, each failure can take down two paths at the same time. For example, failure f_2 fails both Emporia and Pittsburg and failure f_3 fails both Topeka and Emporia. GeoDivRP maintains at least one working path during each failure and the worst case performance for our protocol is at 30 Mb/s.

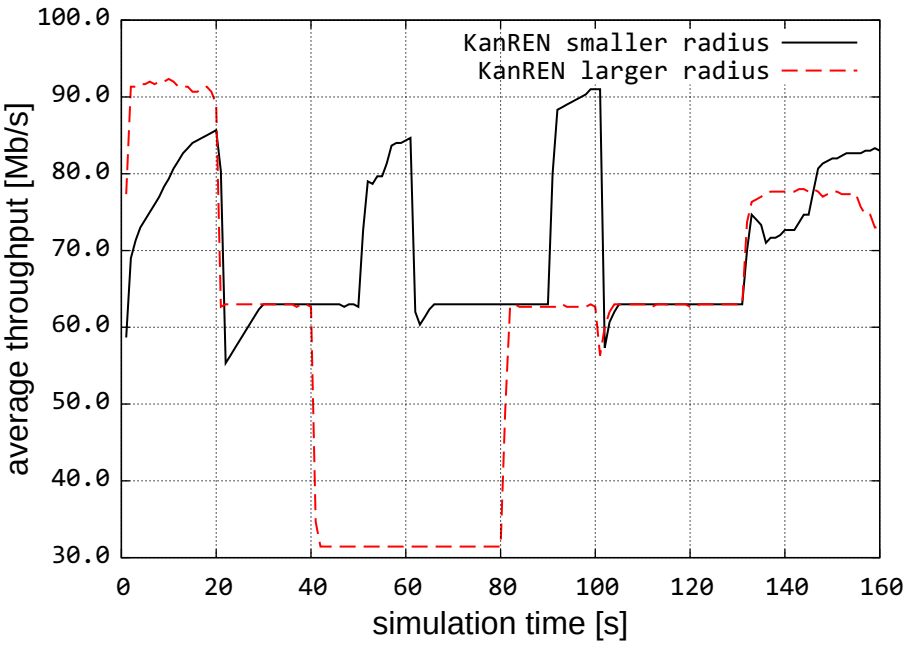

Figure 6.14: KanREN testbed experiment results

Page left intentionally blank.

Chapter 7

Conclusions and Future Work

In this chapter, we conclude this dissertation and propose future work. In Chapter 3, we define *geodiversity* and propose a global graph resilience metric, cTGGD, to represent and compare the geodiversity among different network topologies. A *critical-region identification* mechanism is proposed and verified in identifying critical regions in different topologies; the critical regions further guide the design of resilient protocols. We also analyze how attackers could maximize their attack impact using a fixed budget with a proper knowledge of the network structure; the result helps the design of restoration plans.

In Chapter 4, we further propose two geodiversity heuristics to efficiently solve the path geodiverse problem (PGD): iterative WayPoint Shortest Path (iWPSP) and Modified Link Weight (MLW). We incorporate both of the heuristics in GeoDivRP and demonstrate their effectiveness in calculating and choosing single or multiple GeoPaths to meet the requirements from ResTP and different application scenarios.

Furthermore, in Chapter 5, we incorporate the minimum-cost and the delay-skew requirement in GeoDivRP to solve the flow geodiverse problem (FGD). We generate a linear programming (LP) and a non-linear programming (NLP) formulation to efficiently solve FGD. GeoDivRP chooses either of the formulation and improves the overall link

utilization or delay-skew requirement.

Our ResTP–GeoDivRP protocol stack incorporates the cross-layer information for resilient traffic transmission. We demonstrate our protocol stack's efficiency in bypassing the failure region and its improvement in packet delivery ratio compared to OSPF using UDP, or throughput to MPTCP. When multipath forwarding is considered, our protocol stack performs about 30 – 40% better than MPTCP.

Lastly, we evaluate our cross-layer protocol in the software-defined networking environment in Chapter 6. Our protocol stack takes advantage of the link-layer notification and efficiently responds to network failures by establishing multiple subflows using a set of GeoPaths.

7.1 Future Work

We plan to explore the tradeoff between the multipath and single path routing to understand which one works better in a specific application scenario. For some scenarios, multipath routing can be beneficial in both wired network [70, 110, 112, 123, 135, 176, 177] and wireless network [178–180]. However, recent research has shown that multipath routing is not always beneficial [115]. Furthermore, we plan to examine different load balancing mechanisms to achieve the best routing results under geo-correlated challenges. Our two routing heuristics have different suitable scenarios, a better heuristic is planned to combine the two heuristics into one for calculating GeoPaths.

We plan to incorporate improvement plans for our protocol stack's optimization engine. One possible plan is to provide the GeoPaths to ResTP immediately after calculation and adjust the traffic allocation for ResTP after optimization. Another possible improvement is a distributed algorithm for the delay-skew optimization.

Chapter 8

Acknowledgements

I would like to sincerely thank my advisor, Prof. James P.G. Sterbenz for his continuous support and illuminating instructions during my studies. I thank the committee members: Prof. Victor S. Frost, Prof. Fengjun Li, Prof. Gary J. Minden, Prof. Deep Medhi, Prof. Michael S. Vitevitch and Prof. Jiannong Cao for their valuable feedback to improve this document.

I would like to thank Abdul Jabbar, Justin P. Rohrer, and Egemen Çetinkaya for their help during my early stage in the group. I would like thank Junyan Li for his help on implementing GeoDivRP. I would like to thank Mohammed J.F. Alenazi, Dongsheng Zhang, Siddharth Gangadhar, and Truc Anh N. Nguyen for their suggestion and discussions about research ideas.

I feel very fortunate to work with Dr. Deep Medhi and Todd Gardner from UMKC in the design and evaluation process of GeoDivRP.

I received support during my doctoral studies partially through NSF grant CNS-1128122 (KanREN-GENI), NSF grant CNS-1219028 and CNS-1217736 (Resilient Network Design for Massive Failures and Attacks) at KU and UMKC. I was also supported by teaching assistantship at the Department of Electrical Engineering and Computer

Science, the University of Kansas.

Finally, I am grateful to my family for their never-ending support and inspiration. I will not have gotten this far without their support.

Bibliography

[1] Wikipedia: 2015 Nepal earthquake. `http://en.wikipedia.org/wiki/2015_Nepal_earthquake`, April 2015.

[2] Wikipedia: 2012 Indian Blackouts. `http://en.wikipedia.org/wiki/2012_India_blackouts`, July 2012.

[3] James P. G. Sterbenz, David Hutchison, Egemen K. Çetinkaya, Abdul Jabbar, Justin P. Rohrer, Marcus Schöller, and Paul Smith. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer Networks*, 54(8):1245–1265, 2010.

[4] F. A. Kuipers. An overview of algorithms for network survivability. *CN*, 2012:24:24–24:24, January 2012.

[5] Reuven Cohen, Keren Erez, Daniel ben Avraham, and Shlomo Havlin. Resilience of the Internet to Random Breakdowns. *Phys. Rev. Lett.*, 85:4626–4628, November 2000.

[6] D. Magoni. Tearing down the Internet. *IEEE J.Sel. A. Commun.*, 21(6):949–960, September 2006.

[7] Q. Zhou, L. Gao, R. Liu, and S. Cui. *Network Robustness under Large-Scale Attacks*. SpringerBriefs in Computer Science, 8th edition, 2013.

[8] Amund Kvalbein, Audun Fosselie Hansen, Tarik Cicic, Stein Gjessing, and Olav Lysne. Multiple Routing Configurations for Fast IP Network Recovery. *IEEE Transactions on Networking*, 17(2):473–486, July 2008.

[9] S. Kini, S. Ramasubramanian, A. Kvalbein, and A.F. Hansen. Fast Recovery from Dual Link Failures in IP Networks. In *INFOCOM 2009, IEEE*, pages 1368–1376, April 2009.

[10] Renata Teixeira, Keith Marzullo, Stefan Savage, and Geoffrey M. Voelker. In Search of Path Diversity in ISP Networks. In *Proceedings of the 3rd ACM Internet Measurement Conference (IMC)*, pages 313–318, Miami Beach, FL, 2003.

[11] G. Iannaccone, Chen-Nee Chuah, S. Bhattacharyya, and C. Diot. Feasibility of ip restoration in a tier 1 backbone. *IEEE Network*, 18(2):13–19, Mar 2004.

[12] T. Feyessa and M. Bikdash. Geographically-sensitive network centrality and survivability assessment. In *System Theory (SSST), 2011 IEEE 43rd Southeastern Symposium on*, pages 18–23, March 2011.

[13] Egemen K. Çetinkaya, Mohammed J. F. Alenazi, Justin P. Rohrer, and James P. G. Sterbenz. Topology Connectivity Analysis of Internet Infrastructure Using Graph Spectra. In *Proceedings of the 4th IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM)*, pages 752–758, St. Petersburg, October 2012.

[14] Martin J. Fischer Denise M. B. Masi, Eric E. Smith. Understanding and Mitigating Catastrophic Disruption and Attack. *Sigma Journal*, pages 16–22, September 2010.

[15] Sebastian Neumayer, Gil Zussman, Reuven Cohen, and Eytan Modiano. Assessing the Vulnerability of the Fiber Infrastructure to Disasters. *IEEE/ACM Transactions on Networking*, 19(6):1610–1623, 2011.

[16] Pankaj K. Agarwal, A. Efrat, S.K. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman. Network Vulnerability to Single, Multiple, and Probabilistic Physical Attacks. In *Military Communications Conference, 2010 - MILCOM 2010*, pages 1824–1829, 2010.

[17] M.T. Gardner and C. Beard. Evaluating Geographic Vulnerabilities in Networks. In *IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, pages 1–6, May 2011.

[18] Pierre Francois, Clarence Filsfils, John Evans, and Olivier Bonaventure. Achieving sub-second igp convergence in large ip networks. *SIGCOMM Comput. Commun. Rev.*, 35(3):35–44, July 2005.

[19] Egemen K. Çetinkaya, Dan Broyles, Amit Dandekar, Sripriya Srinivasan, and James P. G. Sterbenz. Modelling Communication Network Challenges for Future Internet Resilience, Survivability, and Disruption Tolerance: A Simulation-Based Approach. *Telecommunication Systems*, 52(2):751–766, 2013.

[20] Egemen K. Çetinkaya, Dan Broyles, Amit Dandekar, Sripriya Srinivasan, and James P. G. Sterbenz. A Comprehensive Framework to Simulate Network Attacks and Challenges. In *Proceedings of the 2nd IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM)*, pages 538–544, Moscow, October 2010.

[21] Justin P. Rohrer, Abdul Jabbar, and James P. G. Sterbenz. Path Diversification: A Multipath Resilience Mechanism. In *Proceedings of the IEEE 7th International Workshop on the Design of Reliable Communication Networks (DRCN)*, pages 343–351, Washington, DC, October 2009.

[22] Justin P. Rohrer, Abdul Jabbar, and James P.G. Sterbenz. Path Diversification for Future Internet End-to-End Resilience and Survivability. *Springer Telecommunication Systems*, 56(1):49–67, May 2014.

[23] Yufei Cheng, M. Todd Gardner, Junyan Li, Rebecca May, Deep Medhi, and James P.G. Sterbenz. Optimised Heuristics for a Geodiverse Routing Protocol. In *Proceedings of the IEEE 10th International Workshop on the Design of Reliable Communication Networks (DRCN)*, pages 1–9, Ghent, Belgium, April 2014.

[24] Egemen K. Çetinkaya, Mohammed J. F. Alenazi, Yufei Cheng, Andrew M. Peck, and James P. G. Sterbenz. On the Fitness of Geographic Graph Generators for Modelling Physical Level Topologies. In *Proceedings of the 5th IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM)*, pages 38–45, Almaty, September 2013.

[25] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824 (Experimental), January 2013.

[26] Christoph Paasch and Sébastien Barré. Multipath TCP in the Linux Kernel. `http://www.multipath-tcp.org`, January 2013.

[27] The ns-3 Network Simulator. `http://www.nsnam.org`, July 2009.

[28] An Instant Virtual Network on your Laptop (or other PC). `http://www.mininet.org`, July 2010.

[29] The Kansas Research and Education Network (KanREN). `http://www.kanren.net/`, December 2009.

[30] OpenOpt Optimization Framework. http://openopt.org/Welcome, October 2007.

[31] Justin P. Rohrer, Mohammed J. F. Alenazi, and James P. G. Sterbenz. ResiliNets Topology Map Viewer. http://www.ittc.ku.edu/resilinets/maps/, January 2011.

[32] James P.G. Sterbenz, Egemen K. Çetinkaya, Mahmood A. Hameed, Abdul Jabbar, Qian Shi, and Justin P. Rohrer. Evaluation of Network Resilience, Survivability, and Disruption Tolerance: Analysis, Topology Generation, Simulation, and Experimentation (invited paper). *Telecommunication Systems*, 52(2):705–736, 2013.

[33] Wei Peng, Zimu Li, Yujing Liux, and Jinshu Su. Assessing the Vulnerability of Network Topologies under Large-Scale Regional Failures. *Communications and Networks, Journal of*, 14(4):451–460, Aug 2012.

[34] Bijan Bassiri and Shahram Shah Heydari. Network survivability in large-scale regional failure scenarios. In *Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering (C3S2E)*, pages 83–87, New York, NY, USA, 2009. ACM.

[35] W. R. Graham R. J. Hermann H. M. Kluepfel R. L. Lawson G. K. Soper L. L. Wood J. S. Foster, E. Gjelde and J. B. Woodard. Report of the Commission to Assess the Threat to the United States from Electromagnetic Pulse (EMP) Attack. Technical report, Critical National Infrastructures, 2008.

[36] Report of the Commission to Assess the Threat to the United States from Electromagnetic Pulse (EMP) Attack. Report, Critical National Infrastructures, 2004.

[37] Clay Wilson. High Altitude Electromagnetic Pulse (HEMP) and High Power Microwave (HPM) Devices: Threat Assessments. DTIC Document, 2008.

[38] Alik Ismail-Zadeh, Jaime Urrutia Fucugauchi, Andrzej Kijko, Kuniyoshi Takeuchi, and Ilya Zaliapin. *Extreme Natural Hazards, Disaster Risks and Societal Implications*, volume 1. Cambridge University Press, 2014.

[39] Egemen K. Çetinkaya and James P. G. Sterbenz. A Taxonomy of Network Challenges. In *Proceedings of the 9th IEEE/IFIP International Conference on the Design of Reliable Communication Networks (DRCN)*, pages 322–330, Budapest, March 2013.

[40] Dongsheng Zhang and James P. G. Sterbenz. Analysis of Critical Node Attacks in Mobile Ad Hoc Networks. In *Proceedings of the 6th IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM)*, pages 171–178, Barcelona, Spain, November 2014.

[41] Petter Holme, Beom Jun Kim, Chang No Yoon, and Seung Kee Han. Attack Vulnerability of Complex Networks. *Phys. Rev. E*, 65:056109, May 2002.

[42] Egemen K. Cetinkaya, Mohammed J.F. Alenazi, Yufei Cheng, Andrew M. Peck, and James P.G. Sterbenz. A comparative analysis of geometric graph models for modelling backbone networks. *Optical Switching and Networking*, 14, Part 2:95 – 106, 2014.

[43] Mohammed J.F. Alenazi, Egemen K. Çetinkaya, and James P. G. Sterbenz. Cost-Constrained and Centrality-Balanced network design improvement. In *RNDM'14 - 6th International Workshop on Reliable Networks Design and Modeling (RNDM 2014)*, pages 194 – 201, Barcelona, Spain, Nov 2014.

[44] Mohammed J.F. Alenazi, Egemen K. Çetinkaya, and James P.G. Sterbenz. Cost-efficient algebraic connectivity optimisation of backbone networks. *Optical Switching and Networking*, 14, Part 2:107 – 116, 2014.

[45] Stephen P. Borgatti. Centrality and Network Flow. *Social Networks*, 27(1):55–71, 2005.

[46] Stephen P. Borgatti and Martin G. Everett. A Graph-Theoretic Perspective on Centrality. *Social Networks*, 28(4):466–484, 2006.

[47] Linton C. Freeman. Centrality in Social Networks Conceptual Clarification. *Social Networks*, 1(3):215–239, 1978–1979.

[48] Linton C. Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1):35–41, 1977.

[49] Mark EJ Newman. The Mathematics of Networks. *The new palgrave encyclopedia of economics*, 2:1–12, 2008.

[50] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring ISP Topologies with Rocketfuel. *IEEE/ACM Transactions on Networking*, 12(1):2–16, 2004.

[51] Ramakrishnan Durairajan, Paul Barford, Joel Sommers, and Walter Willinger. Intertubes: A study of the us long-haul fiber-optic infrastructure. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 565–578, New York, NY, USA, 2015. ACM.

[52] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0–Survivable Network Design Library. *Networks*, 55(3):276–286, 2010.

[53] Simon Knight, Hung X. Nguyen, Nickolas Falkner, Rhys Bowden, and Matthew Roughan. The Internet Topology Zoo. *IEEE Journal on Selected Areas in Communications*, 29(9):1765–1775, 2011.

[54] Austin Cosner, Justin P. Rohrer, Mohammed J. F. Alenazi, and James P. G. Sterbenz. ResiliNets Topology Map Viewer Version 2. `http://www.ittc.ku.edu/resilinets/maps/v2`, January 2015.

[55] James P. G. Sterbenz and David Hutchison. Resilinets: Multilevel resilient and survivable networking initiative wiki. `http://wiki.ittc.ku.edu/resilinets`, April 2006.

[56] K. Ruben Gabriel and Robert R. Sokal. A New Statistical Approach to Geographic Variation Analysis. *Systematic Zoology*, 18(3):259–278, 1969.

[57] David W. Matula and Robert R. Sokal. Properties of Gabriel Graphs Relevant to Geographic Variation Research and the Clustering of Points in the Plane. *Geographical Analysis*, 12(3):205–222, 1980.

[58] Bernard M. Waxman. Routing of Multipoint Connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.

[59] J. W. Suurballe. Disjoint Paths in a Network. *Networks*, 4(2):125–145, 1974.

[60] J. W. Suurballe and R. E. Tarjan. A Quick Method for Finding Shortest Pairs of Disjoint Paths. *Networks*, 14(2):325–336, 1984.

[61] Ramesh Bhandari. Optimal Diverse Routing in Telecommunication Fiber Networks. In *Proceedings of the IEEE INFOCOM*, volume 3, pages 1498–1508, Toronto, June 1994.

[62] Jiayue He and Jennifer Rexford. Toward Internet-Wide Multipath Routing. *IEEE Network Magazine*, 22(2):16–21, 2008.

[63] Murtaza Motiwala, Megan Elmore, Nick Feamster, and Santosh Vempala. Path Splicing. In *Proceedings of the ACM SIGCOMM*, pages 27–38, Seattle, WA, August 2008.

[64] Xiaowei Yang and David Wetherall. Source Selectable Path Diversity via Routing Deflections. In *Proceedings of the ACM SIGCOMM*, pages 159–170, Pisa, September 2006.

[65] Ariel Orda and Alexander Sprintson. Efficient Algorithms for Computing Disjoint QoS Paths. In *Proceedings of the IEEE INFOCOM*, volume 1, pages 727–738, Hong Kong, March 2004.

[66] Yuchun Guo, Fernando Kuipers, and Piet Van Mieghem. Link-Disjoint Paths for Reliable QoS Routing. *International Journal of Communication Systems*, 16(9):779–798, 2003.

[67] Feng Wang and Lixin Gao. Path Diversity Aware Interdomain Routing. In *Proceedings of the IEEE INFOCOM*, pages 307–315, Rio de Janeiro, April 2009.

[68] Hyang-Won Lee, E. Modiano, and Kayi Lee. Diverse Routing in Networks With Probabilistic Failures. *IEEE/ACM Transactions on Networking*, 18(6):1895–1907, 2010.

[69] Aun Haider and Richard Harris. Recovery Techniques in Next Generation Networks. *IEEE Communications Surveys & Tutorials*, 9(3):2–17, 2007.

[70] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley. Multi-Path TCP: A Joint Congestion Control and Routing Scheme to Exploit Path Diversity in the Internet. *IEEE/ACM Transactions on Networking*, 14(6):1260–1271, 2006.

[71] P. Babarczi, J. Tapolcai, Pin-Han Ho, and M. Medard. Optimal Dedicated Protection Approach to Shared Risk Link Group Failures using Network Coding. In *Communications (ICC), 2012 IEEE International Conference on*, pages 3051–3055, June 2012.

[72] Dimitri Papadimitriou, Fabrice Poppe, Jim Jones, Senthil Venkatachalam, Sudheer Dharanikota, Raj Jain, Riad Hartani, David Griffith, and Y Xue. Inference of Shared Risk Link Groups. Internet Draft, May 2002.

[73] Ramesh Bhandari. *Survivable Networks: Algorithms for Diverse Routing*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.

[74] J. Armitage, O. Crochat, and J.Y. Le Boudec. Design of a Survivable WDM Photonic Network. In *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*, volume 1, pages 244–252 vol.1, Apr 1997.

[75] O. Crochat and J.-Y. Le Boudec. Design Protection for WDM Optical Networks. *Selected Areas in Communications, IEEE Journal on*, 16(7):1158–1165, Sep 1998.

[76] Olivier Crochat, Jean-Yves Le Boudec, and Ornan Gerstel. Protection Interoperability for WDM Optical Networks. *IEEE/ACM Trans. Netw.*, 8(3):384–395, June 2000.

[77] E. Modiano and A. Narula-Tam. Survivable Lightpath Routing: a New Approach to the Design of WDM-Based Networks. *Selected Areas in Communications, IEEE Journal on*, 20(4):800–809, May 2002.

[78] Jian Qiang Hu. Diverse routing in optical mesh networks. *Communications, IEEE Transactions on*, 51(3):489–494, March 2003.

[79] A. Narula-Tam, E. Modiano, and A. Brzezinski. Physical Topology Design for Survivable Routing of Logical Rings in WDM-Based Networks. *Selected Areas in Communications, IEEE Journal on*, 22(8):1525–1538, Oct 2004.

[80] Pallab Datta and Arun K Somani. Graph transformation approaches for diverse routing in shared risk resource group (srrg) failures. *Computer Networks*, 52(12):2381–2394, 2008.

[81] James P. G. Sterbenz, David Hutchison, Egemen K. Çetinkaya, Abdul Jabbar, Justin P. Rohrer, Marcus Schöller, and Paul Smith. Redundancy, Diversity, and Connectivity to Achieve Multilevel Network Resilience, Survivability, and Disruption Tolerance (invited paper). *Telecommunication Systems*, 56(1):17–31, 2014.

[82] M.F. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, and B. Mukherjee. A Disaster-Resilient Multi-Content Optical Datacenter Network Architecture. In *Transparent Optical Networks (ICTON), 2011 13th International Conference on*, pages 1–4, June 2011.

[83] Reuven Cohen, Keren Erez, Daniel Ben-Avraham, and Shlomo Havlin. Breakdown of the Internet under Intentional Attack. *Physical review letters*, 86(16):3682, 2001.

[84] S. Neumayer and E. Modiano. Network Reliability With Geographically Correlated Failures. In *Proc. of IEEE INFOCOM*, pages 1–9, March 2010.

[85] W. Wu, B. Moran, J.H. Manton, and M. Zukerman. Topology Design of Undersea Cables Considering Survivability Under Major Disasters. In *International Conference on WAINA*, pages 1154–1159, May 2009.

[86] M.T. Gardner, C. Beard, and D. Medhi. 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, pages 264–271, March 2012.

[87] A. Sen, Bao Hong Shen, Ling Zhou, and Bin Hao. Fault-Tolerance in Sensor Networks: A New Evaluation Metric. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006.

[88] A. Sen, S. Murthy, and S. Banerjee. Region-Based Connectivity - a New Paradigm for Design of Fault-Tolerant Networks. In *High Performance Switching and Routing, 2009. HPSR 2009. International Conference on*, pages 1–7, June 2009.

[89] Jacek Rak. Measures of Region Failure Survivability for Wireless Mesh Networks. *Wireless Networks*, 21(2):673–684, 2015.

[90] Richard L. Church, Maria P. Scaparra, and Richard S. Middleton. Identifying Critical Infrastructure: The Median and Covering Facility Interdiction Problems. *Annals of the Association of American Geographers*, 94(3):491–502, 2004.

[91] Béla Bollobás. The evolution of random graphs. *Transactions of the American Mathematical Society*, 286(1):257–274, 1984.

[92] R. Barlow and F. Proschan. *Mathematical Theory of Reliability*. Society for Industrial and Applied Mathematics, 1996.

[93] Vito Latora and Massimo Marchiori. Efficient behavior of small-world networks. *Phys. Rev. Lett.*, 87:198701, Oct 2001.

[94] M. Rahnamay-Naeini, J.E. Pezoa, G. Azar, N. Ghani, and M.M. Hayat. Modeling Stochastic Correlated Failures and their Effects on Network Reliability. In *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, pages 1–6, July 2011.

[95] David J Strauss. A Model for Clustering. *Biometrika*, 62(2):467–475, 1975.

[96] M.T. Gardner, C. Beard, and D. Medhi. Using Network Measure to Reduce State Space Enumeration in Resilient Networks. In *Design of Reliable Communication Networks (DRCN), 2013 9th International Conference on the*, pages 250–257, 2013.

[97] H. Saito. Analysis of geometric disaster evaluation model for physical networks. *Networking, IEEE/ACM Transactions on*, PP(99):1–1, 2014.

[98] James Joseph Sylvester. A question in the geometry of situation. *Quarterly Journal of Pure and Applied Mathematics*, 1, 1857.

[99] P. K. Agarwal, A. Efrat, S. K. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman. The Resilience of WDM Networks to Probabilistic Geographical Failures. *IEEE/ACM Transactions on Networking*, PP(99):1, 2013.

[100] Stojan Trajanovski, Fernando A Kuipers, Aleksandar Ilic, Jon Crowcroft, and Piet Van Mieghem. Finding critical regions and region-disjoint paths in a network. *Networking, IEEE/ACM Transactions on*, PP(99):1–1, 2014.

[101] E. W. Dijkstra. A Note on Two Problems in connection with Graphs. *Numerische Mathematik*, 1:269–271, 1959.

[102] Renata Teixeira, Keith Marzullo, Stefan Savage, and Geoffrey M. Voelker. Characterizing and Measuring Path Diversity of Internet Topologies. *SIGMETRICS Perform. Eval. Rev.*, 31(1):304–305, June 2003.

[103] Yufei Cheng, M. Todd Gardner, Junyan Li, Rebecca May, Deep Medhi, and James P.G. Sterbenz. Analysing GeoPath Diversity and Improving Routing Performance in Optical Networks. *Computer Networks*, 82:50–67, May 2015.

[104] C. Hopps. Analysis of an Equal-Cost Multi-Path Algorithm. RFC 2992 (Informational), November 2000.

[105] Mike Shand and Stewart Bryant. IP Fast Reroute Framework. RFC 5714, October 2015.

[106] Jin Y Yen. Finding the k Shortest Loopless Paths in a Network. *management Science*, 17(11):712–716, 1971.

[107] Srikanth Kandula, Dina Katabi, Bruce Davie, and Anna Charny. Walking the Tightrope: Responsive Yet Stable Traffic Engineering. In *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '05, pages 253–264, New York, NY, USA, 2005. ACM.

[108] Meng Wang, Chee Wei Tan, Weiyu Xu, and Ao Tang. Cost of Not Splitting in Routing: Characterization and Estimation. *IEEE/ACM Trans. Netw.*, 19(6):1849–1859, December 2011.

[109] Charu C. Aggarwal and James B. Orlin. On Multiroute Maximum Flows in Networks. *Networks*, 39(1):43–52, 2002.

[110] Bao Hong Shen, Bin Hao, and A. Sen. On Multipath Routing using Widest Pair of Disjoint Paths. In *High Performance Switching and Routing, 2004. HPSR. 2004 Workshop on*, pages 134–140, 2004.

[111] Mateusz Dzida, Michal Zagozdzon, Mateusz Zotkiewicz, and Michal Pioro. Flow Optimization in IP Networks with Fast Proactive Recovery. In *Telecommunications Network Strategy and Planning Symposium, 2008. Networks 2008. The 13th International*, volume Supplement, pages 1–9, Sept 2008.

[112] Junjie Zhang, Kang Xi, L. Zhang, and H.J. Chao. Optimizing Network Performance Using Weighted Multipath Routing. In *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, pages 1–7, July 2012.

[113] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient Overlay Networks. In *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, SOSP '01, pages 131–145, New York, NY, USA, 2001. ACM.

[114] Stefan Savage, Tom Anderson; Amit Aggarawl, Tom Anderson, Amit Aggarwal, David Becker, Neal Cardwell, Andy Collins, Eric Hoffman, John Snell, Amin Vahdat, Geoff Voelker, and John Zahorjan. Detour: a Case for Informed Internet Routing and Transport. *IEEE Micro*, 19:50–59, 1999.

[115] Xuan Liu, S. Mohanraj, M. Pioro, and D. Medhi. Multipath Routing from a Traffic Engineering Perspective: How Beneficial Is It? In *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, pages 143–154, Oct 2014.

[116] M.T. Gardner, R. May, C. Beard, and D. Medhi. Using Multi-Topology Routing to Improve Routing during Geographically Correlated Failures. In *Design of Reliable Communication Networks (DRCN), 2014 10th International Conference on the*, pages 1–8, April 2014.

[117] D. Xu, M. Chiang, and J. Rexford. Deft: Distributed exponentially-weighted flow splitting. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pages 71–79, May 2007.

[118] H Tahilramani Kaur, Shivkumar Kalyanaraman, Andreas Weiss, Shifalika Kanwar, and Ayesha Gandhi. BANANAS: An Evolutionary Framework for Explicit and Multipath Routing in the Internet. *ACM SIGCOMM Computer Communication Review*, 33(4):277–288, 2003.

[119] Srikanth Kandula, Dina Katabi, Shantanu Sinha, and Arthur Berger. Dynamic load balancing without packet reordering. *SIGCOMM Comput. Commun. Rev.*, 37(2):51–62, March 2007.

[120] R. Banner and A.. Orda. Multipath Routing Algorithms for Congestion Minimization. *Networking, IEEE/ACM Transactions on*, 15(2):413–424, April 2007.

[121] Michael Menth, Rüdiger Martin, Arie MCA Koster, and Sebastian Orlowski. Overview of Resilience Mechanisms Based on Multipath Structures. In *Design and Reliable Communication Networks, 2007. DRCN 2007. 6th International Workshop on*, pages 1–9. IEEE, 2007.

[122] A Iselt, A Kirstadter, A Pardigon, and Thomas Schwabe. Resilient Routing using MPLS and ECMP. In *High Performance Switching and Routing, 2004. HPSR. 2004 Workshop on*, pages 345–349. IEEE, 2004.

[123] Srihari Nelakuditi and Zhi-Li Zhang. On Selection of Paths for Multipath Routing. In *Proceedings of the 9th International Workshop on Quality of Service*, IWQoS '01, pages 170–186, London, UK, UK, 2001. Springer-Verlag.

[124] T. Anjali, A. Fortin, G. Calinescu, S. Kapoor, N. Kirubanandan, and S. Tongngam. Multipath Network Flows: Bounded Buffers and Jitter. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–7, March 2010.

[125] T. Ishida, K. Ueda, and T. Yakoh. Fairness and Utilization in Multipath Network Flow Optimization. In *Industrial Informatics, 2006 IEEE International Conference on*, pages 1096–1101, Aug 2006.

[126] Ming Zhang, Junwen Lai, Arvind Krishnamurthy, Larry Peterson, and Randolph Wang. A transport layer approach for improving end-to-end performance and ro-

bustness using redundant paths. In *Proceedings of the USENIX Annual Tchnical Conference*, Boston, MA, June 2004.

[127] J.R. Iyengar, P.D. Amer, and R. Stewart. Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-End Paths. *Networking, IEEE/ACM Transactions on*, 14(5):951–964, Oct 2006.

[128] Truc Anh N. Nguyen, Justin P. Rohrer, and James P. G. Sterbenz. ResTP–A Transport Protocol for FI Resilience. In *Proceedings of the 10th International Conference on Future Internet Technologies (CFI)*, June 2015.

[129] Justin P. Rohrer, Ramya Naidu, and James P. G. Sterbenz. Multipath at the Transport Layer: An End-to-End Resilience Mechanism. In *Proceedings of the IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM)*, pages 1–7, St. Petersburg, Russia, October 2009.

[130] David Feldmeier. An overview of the TP++ transport protocol project. In Ahmed N. Tantawy, editor, *High Performance Networks: Frontiers and Experience*, volume 238 of *Kluwer International Series in Engineering and Computer Science*, chapter 8. Kluwer Academic Publishers, Boston, MA, USA, 1993.

[131] Dimitri Bertsekas and Robert Gallager. *Data Networks (2Nd Ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[132] Bruce Davie and Yakov Rekhter. *MPLS: Technology and Applications*. Morgan Kaufmann Publishers Inc., 2000.

[133] Biswanath Mukherjee. *Optical Communication Networks*. McGraw-Hill Companies, 1997.

[134] S. Mostafa Mostafavi, Ehsan Hamadani, and Rahim Tafazolli. Delay Minimization in Multipath Routing. In *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, IWCMC '10, pages 711–715, New York, NY, USA, 2010. ACM.

[135] Fabrizio Devetak, Junghwan Shin, Tricha Anjali, and Sanjiv Kapoor. Minimizing Path Delay in Multipath Networks. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[136] Srinivas Vutukury and JJ Garcia-Luna-Aceves. *A Simple Approximation to Minimum-Delay Routing*, volume 29. ACM, 1999.

[137] H. Dahmouni, A. Girard, M. Ouzineb, and B. Sanso. The Impact of Jitter on Traffic Flow Optimization in Communication Networks. *Network and Service Management, IEEE Transactions on*, 9(3):279–292, September 2012.

[138] Michal Pióro and Deepankar Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[139] Cynthia A. Phillips. The Network Inhibition Problem. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 776–785, New York, NY, USA, 1993. ACM.

[140] Carl Burch, Robert Carr, Sven Krumke, Madhav Marathe, Cynthia Phillips, and Eric Sundberg. A Decomposition-Based Pseudoapproximation Algorithm for Network Flow Inhibition. In *Network Interdiction and Stochastic Integer Programming*, pages 51–68. Springer, 2003.

[141] S Alexander. On the History of Combinatorial Optimization (till 1960). *Handbooks in Operations Research and Management Science: Discrete Optimization*, 12:1, 2005.

[142] Ali Pinar, Yonatan Fogel, and Bernard Lesieutre. The Inhibiting Bisection Problem. *Lawrence Berkeley National Laboratory*, 2006.

[143] Deepankar Medhi and Karthikeyan Ramasamy. *Network Routing: Algorithms, Protocols, and Architectures*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.

[144] Mayur Channegowda, Reza Nejabati, and Dimitra Simeonidou. Software-Defined Optical Networks Technology and Infrastructure: Enabling Software-Defined Optical Network Operations [Invited]. *Journal of Optical Communications and Networking*, 5(10):A274–A282, 2013.

[145] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: Enabling Innovation in Campus Networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, March 2008.

[146] Marcel Großmann and Stephan JA Schuberth. Auto-Mininet: Assessing the Internet Topology Zoo in a Software-Defined Network Emulator.

[147] An Xie, Xiaoliang Wang, Wei Wang, and Sanglu Lu. Designing a Disaster-Resilient Network with Software Defined Networking. In *Quality of Service (IWQoS), 2014 IEEE 22nd International Symposium of*, pages 135–140, May 2014.

[148] Justin P. Rohrer and James P. G. Sterbenz. Predicting topology survivability using path diversity. In *Proceedings of the IEEE/IFIP International Workshop on*

Reliable Networks Design and Modeling (RNDM), pages 95–101, Budapest, October 2011.

[149] Charles J. Colbourn and Daryl D. Harms. Bounding All-terminal Reliability in Computer Networks. *Networks*, 18(1):1–12, 1988.

[150] D Jack Elzinga and Donald W Hearn. The minimum covering sphere problem. *Management science*, 19(1):96–104, 1972.

[151] Level 3 Network Map. `http://maps.level3.com`.

[152] Sprint. `http://www.sprint.com`.

[153] Bestel. `http://www.bestel.com.mx/`.

[154] OTEGLOBE Network Map. `http://www.oteglobe.gr/`.

[155] euNetworks. `http://www.eunetworks.com/`.

[156] NORDUnet: Nordic infrastructure for research & education. `http://www.nordu.net/`, December 2009.

[157] AT&T. `http://www.att.com`.

[158] Internet2. `http://www.internet2.edu`.

[159] TeliaSonera. `http://www.teliasoneraic.com`.

[160] The Next Generation Core Optical Networks (CORONET). `http://www.darpa.mil/Our_Work/STO/Programs/Dynamic_Multi-Terabit_Core_Optical_Networks_(CORONET).aspx`.

[161] Swedish University Computer Network Map. `http://basun.sunet.se/engelska.html`.

[162] A. Sahoo, K. Kant, and P. Mohapatra. Characterization of BGP Recovery Time under Large-Scale Failures. In *IEEE ICC*, volume 2, pages 949–954, June 2006.

[163] Eric Bouillet, Georgios Ellinas, Jean-François Labourdette, and Ramu Ramamurthy. *Optical Networking*, pages 1–23. John Wiley & Sons, Ltd, 2007.

[164] M Todd Gardner, Rebecca May, Cory Beard, and Deep Medhi. A Geographic Multi-Topology Routing approach and its Benefits during Large-Scale Geographically Correlated Failures. *Computer Networks*, 2015.

[165] Vedat Akgün, Erhan Erkut, and Rajan Batta. On Finding Dissimilar Paths. *European Journal of Operational Research*, 121(2):232–246, 2000.

[166] J. Moy. OSPF specification. RFC 1131 (Proposed Standard), October 1989. Obsoleted by RFC 1247.

[167] Justin P. Rohrer, Abdul Jabbar, Egemen K. Çetinkaya, and James P.G. Sterbenz. Airborne Telemetry Networks: Challenges and Solutions in the ANTP Suite. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, pages 74–79, San Jose, CA, November 2010.

[168] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for Traffic Engineering Over MPLS. RFC 2702 (Informational), September 1999.

[169] Leonard Kleinrock. *Queueing Systems*, volume II: Computer Applications. Wiley Interscience, 1976. (Published in Russian, 1979. Published in Japanese, 1979.).

[170] David Gómez Fernández. Multipath tcp in ns-3. `https://github.com/dgomezunican/multipath-ns3.13`, 2013.

[171] GENI: Global environment for network innovations. `http://www.geni.net/`, December 2009.

[172] http://www.brocade.com/, 1995.

[173] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031 (Proposed Standard), January 2001.

[174] The Floodlight Controller. http://www.projectfloodlight.org/, 2015.

[175] iPerf - The network bandwidth measurement tool. https://iperf.fr/, March 2003.

[176] P. Merindol, J.-J. Pansiot, and S. Cateloin. Improving load balancing with multipath routing. In *Proceedings of 17th International Conference on Computer Communications and Networks, 2008. ICCCN '08.*, pages 1–8, August 2008.

[177] Igor Ganichev, Bin Dai, P Godfrey, and Scott Shenker. YAMR: Yet Another Multipath Routing Protocol. *ACM SIGCOMM Computer Communication Review*, 40(5):13–19, 2010.

[178] Sung-Ju Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 10, pages 3201–3205 vol.10, 2001.

[179] Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, and Robert Morris. Performance of multihop wireless networks: Shortest path is not enough. In *Proceedings of the First Workshop on Hot Topics in Networks (HotNets-I)*, Princeton, New Jersey, October 2002. ACM SIGCOMM.

[180] Asis Nasipuri, Robert Castañeda, and Samir R. Das. Performance of Multipath Routing for On-demand Protocols in Mobile Ad Hoc Networks. *Mob. Netw. Appl.*, 6(4):339–349, August 2001.

Appendix A

Plots for Additional Scenarios

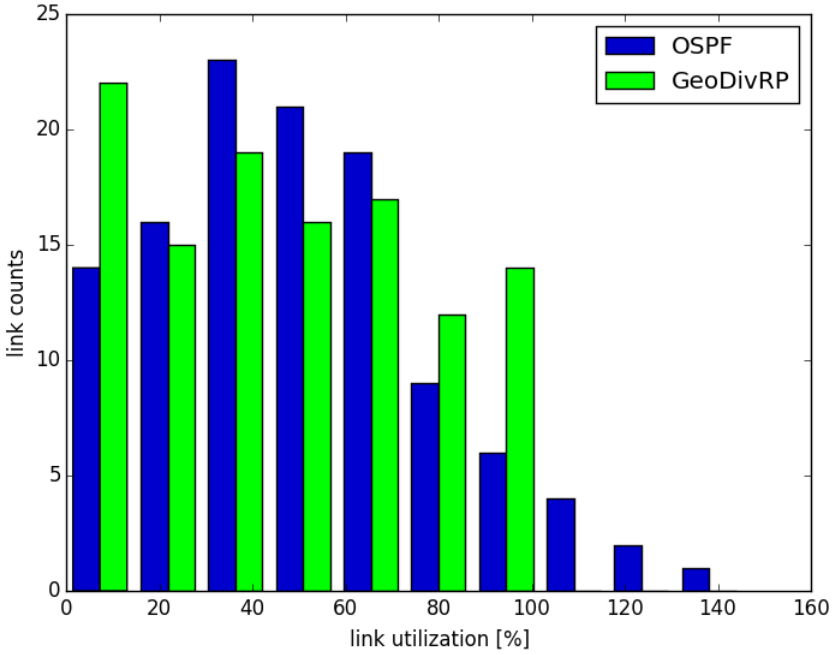

Figure A.1: CORONET network link utilization

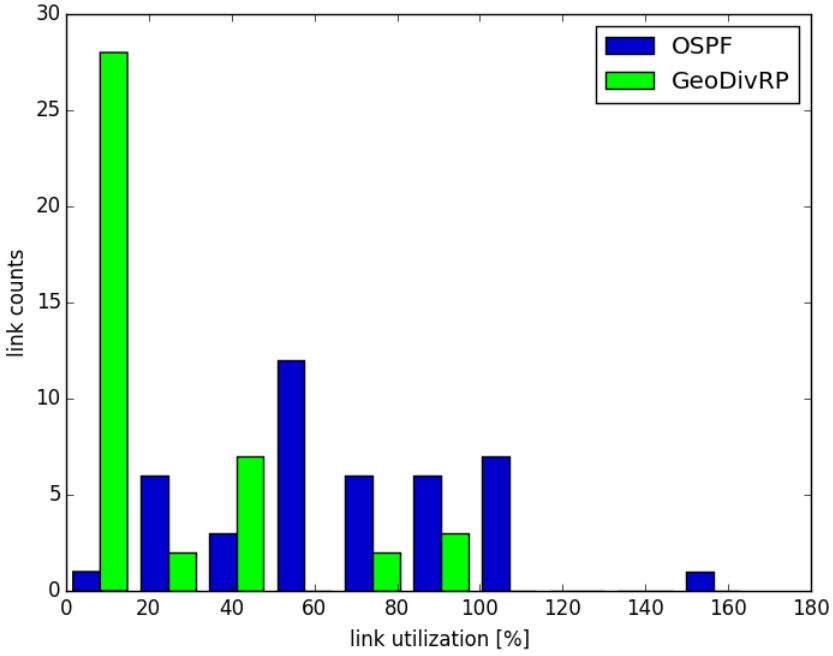

Figure A.2: Internet2 network link utilization

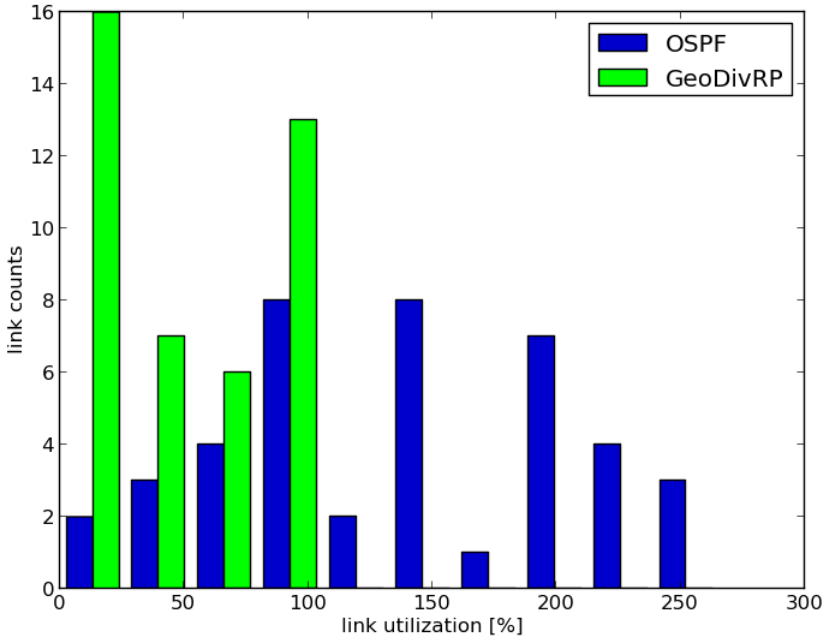

Figure A.3: TeliaSonera network link utilization

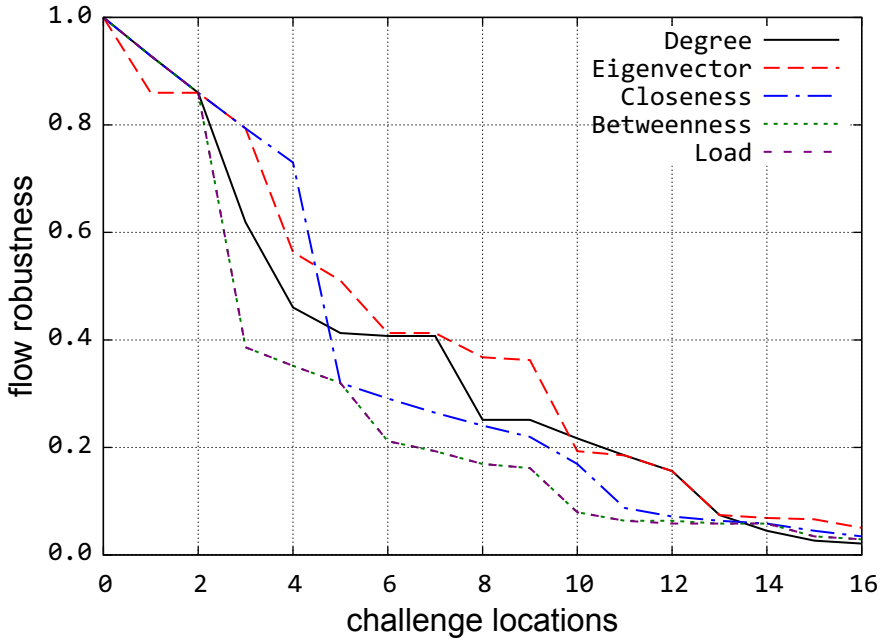

Figure A.4: Nobel optical network under regional challenges

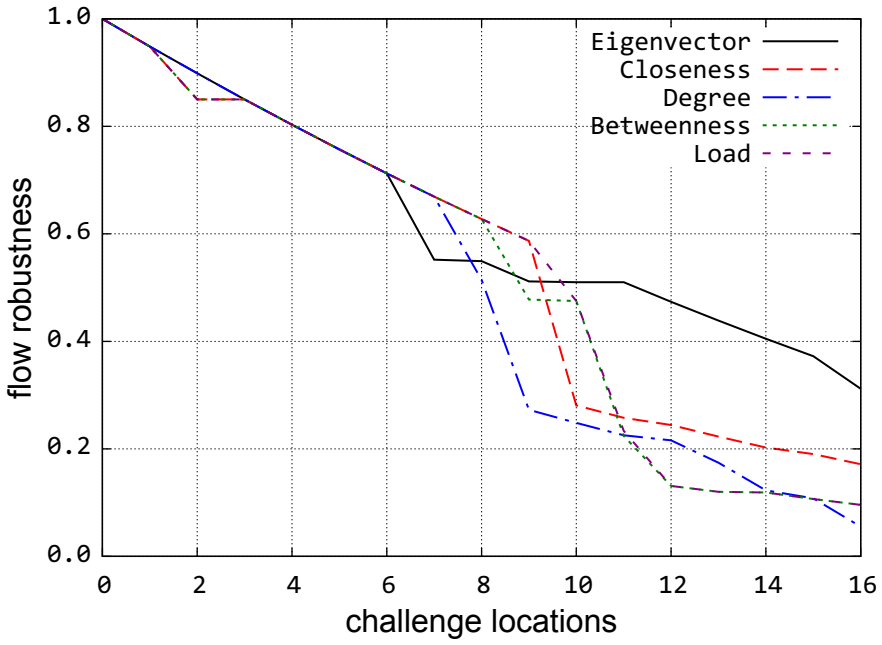

Figure A.5: CORONET optical network under regional challenges

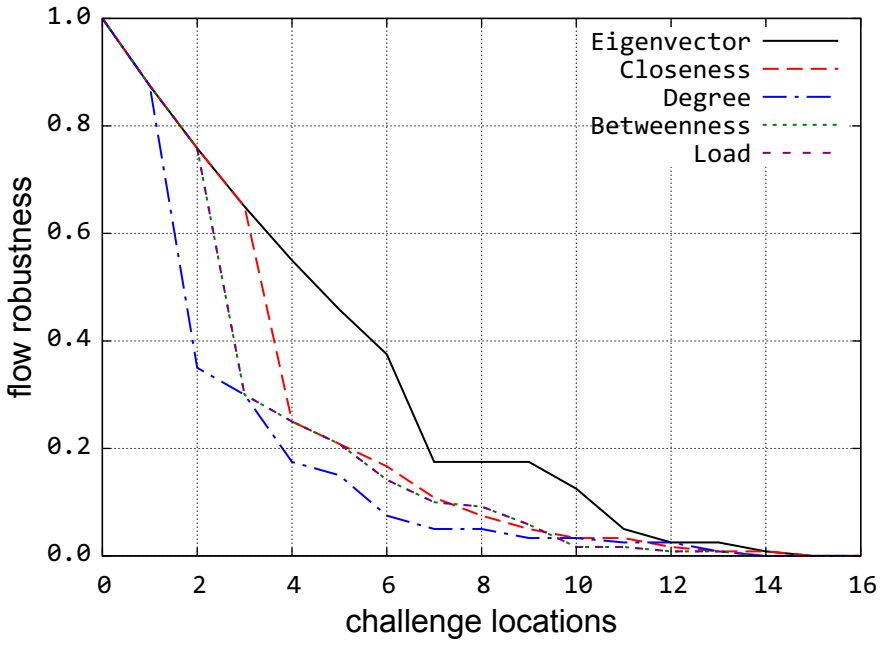

Figure A.6: Internet2 optical network under regional challenges

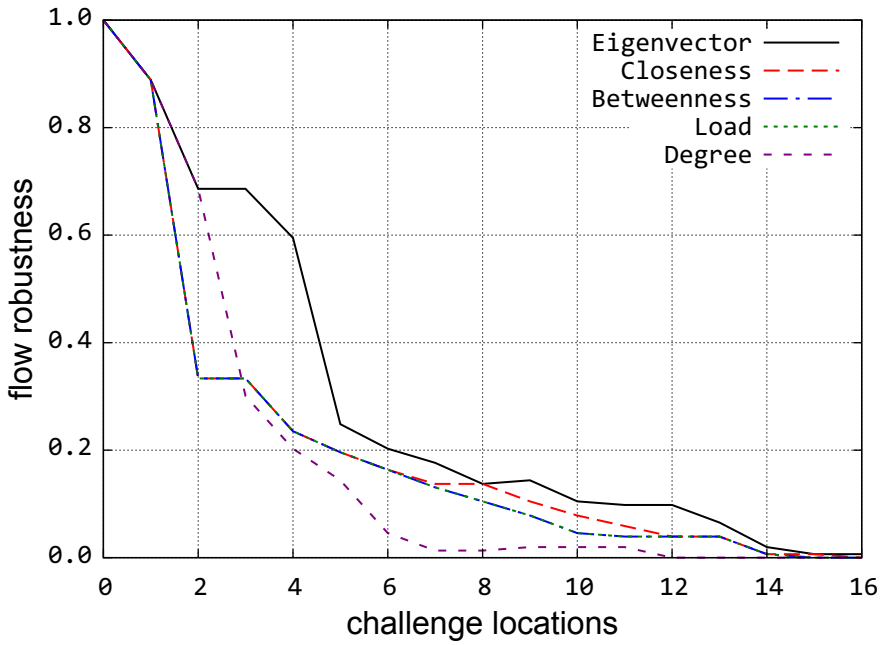

Figure A.7: TeliaSonera optical network under regional challenges