

THE EVALUATION OF SOFTWARE DEFINED NETWORKING
FOR COMMUNICATION AND CONTROL OF CYBER
PHYSICAL SYSTEMS

by

ALI SYDNEY

B.S., United States Naval Academy, 2007

M.S., Kansas State University, 2009

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering
College of Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

2013

Abstract

Cyber physical systems emerge when physical systems are integrated with communication networks. In particular, communication networks facilitate dissemination of data among components of physical systems to meet key requirements, such as efficiency and reliability, in achieving an objective. In this dissertation, we consider one of the most important cyber physical systems: the smart grid.

The North American Electric Reliability Corporation (NERC) envisions a smart grid that aggressively explores advance communication network solutions to facilitate real-time monitoring and dynamic control of the bulk electric power system. At the distribution level, the smart grid integrates renewable generation and energy storage mechanisms to improve reliability of the grid. Furthermore, dynamic pricing and demand management provide customers an avenue to interact with the power system to determine electricity usage that satisfies their lifestyle. At the transmission level, efficient communication and a highly automated architecture provide visibility in the power system; hence, faults are mitigated faster than they can propagate. However, higher levels of reliability and efficiency rely on the supporting physical communication infrastructure and the network technologies employed.

Conventionally, the topology of the communication network tends to be identical to that of the power network. In this dissertation, however, we employ a Demand Response (DR) application to illustrate that a topology that may be ideal for the power network may not necessarily be ideal for the communication network. To develop this illustration, we realize that communication network issues, such as congestion, are addressed by protocols, middle-ware, and software mechanisms. Additionally, a network whose physical topology is designed to avoid congestion realizes an even higher level of performance. For this reason, characterizing the communication infrastructure of smart grids provides mechanisms to improve performance while minimizing cost. Most recently,

algebraic connectivity has been used in the ongoing research effort characterizing the robustness of networks to failures and attacks. Therefore, we first derive analytical methods for increasing algebraic connectivity and validate these methods numerically. Secondly, we investigate impact on the topology and traffic characteristics as algebraic connectivity is increased. Finally, we construct a DR application to demonstrate how concepts from graph theory can dramatically improve the performance of a communication network. With a hybrid simulation of both power and communication network, we illustrate that a topology which may be ideal for the power network may not necessarily be ideal for the communication network.

To date, utility companies are embracing network technologies such as Multiprotocol Label Switching (MPLS) because of the available support for legacy devices, traffic engineering, and virtual private networks (VPNs) which are essential to the functioning of the smart grid. Furthermore, this particular network technology meets the requirement of non-routability as stipulated by NERC, but these benefits are costly for the infrastructure that supports the full MPLS specification. More importantly, with MPLS routing and other switching technologies, innovation is restricted to the features provided by the equipment. In particular, no practical method exists for utility consultants or researchers to test new ideas, such as alternatives to IP or MPLS, on a realistic scale in order to obtain the experience and confidence necessary for real-world deployments. As a result, novel ideas remain untested. On the contrary, OpenFlow, which has gained support from network providers such as Microsoft and Google and equipment vendors such as NEC and Cisco, provides the programmability and flexibility necessary to enable innovation in next-generation communication architectures for the smart grid. This level of flexibility allows OpenFlow to provide all features of MPLS and allows OpenFlow devices to co-exist with existing MPLS devices. Therefore, in this dissertation we explore a low-cost OpenFlow Software Defined Networking solution and compare its performance to that of MPLS.

In summary, we develop methods for designing robust networks and evaluate software defined networking for communication and control in cyber physical systems where the smart grid is the system under consideration.

THE EVALUATION OF SOFTWARE DEFINED NETWORKING
FOR COMMUNICATION AND CONTROL OF CYBER
PHYSICAL SYSTEMS

by

ALI SYDNEY

B.S., United States Naval Academy, 2007

M.S., Kansas State University, 2009

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering
College of Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

2013

Approved by:

Co-Major Professor
Caterina Scoglio

Approved by:

Co-Major Professor
Don Gruenbacher

Copyright

Ali Sydney

2013

Abstract

Cyber physical systems emerge when physical systems are integrated with communication networks. In particular, communication networks facilitate dissemination of data among components of physical systems to meet key requirements, such as efficiency and reliability, in achieving an objective. In this dissertation, we consider one of the most important cyber physical systems: the smart grid.

The North American Electric Reliability Corporation (NERC) envisions a smart grid that aggressively explores advance communication network solutions to facilitate real-time monitoring and dynamic control of the bulk electric power system. At the distribution level, the smart grid integrates renewable generation and energy storage mechanisms to improve reliability of the grid. Furthermore, dynamic pricing and demand management provide customers an avenue to interact with the power system to determine electricity usage that satisfies their lifestyle. At the transmission level, efficient communication and a highly automated architecture provide visibility in the power system; hence, faults are mitigated faster than they can propagate. However, higher levels of reliability and efficiency rely on the supporting physical communication infrastructure and the network technologies employed.

Conventionally, the topology of the communication network tends to be identical to that of the power network. In this dissertation, however, we employ a Demand Response (DR) application to illustrate that a topology that may be ideal for the power network may not necessarily be ideal for the communication network. To develop this illustration, we realize that communication network issues, such as congestion, are addressed by protocols, middle-ware, and software mechanisms. Additionally, a network whose physical topology is designed to avoid congestion realizes an even higher level of performance. For this reason, characterizing the communication infrastructure of smart grids provides mechanisms to improve performance while minimizing cost. Most recently,

algebraic connectivity has been used in the ongoing research effort characterizing the robustness of networks to failures and attacks. Therefore, we first derive analytical methods for increasing algebraic connectivity and validate these methods numerically. Secondly, we investigate impact on the topology and traffic characteristics as algebraic connectivity is increased. Finally, we construct a DR application to demonstrate how concepts from graph theory can dramatically improve the performance of a communication network. With a hybrid simulation of both power and communication network, we illustrate that a topology which may be ideal for the power network may not necessarily be ideal for the communication network.

To date, utility companies are embracing network technologies such as Multiprotocol Label Switching (MPLS) because of the available support for legacy devices, traffic engineering, and virtual private networks (VPNs) which are essential to the functioning of the smart grid. Furthermore, this particular network technology meets the requirement of non-routability as stipulated by NERC, but these benefits are costly for the infrastructure that supports the full MPLS specification. More importantly, with MPLS routing and other switching technologies, innovation is restricted to the features provided by the equipment. In particular, no practical method exists for utility consultants or researchers to test new ideas, such as alternatives to IP or MPLS, on a realistic scale in order to obtain the experience and confidence necessary for real-world deployments. As a result, novel ideas remain untested. On the contrary, OpenFlow, which has gained support from network providers such as Microsoft and Google and equipment vendors such as NEC and Cisco, provides the programmability and flexibility necessary to enable innovation in next-generation communication architectures for the smart grid. This level of flexibility allows OpenFlow to provide all features of MPLS and allows OpenFlow devices to co-exist with existing MPLS devices. Therefore, in this dissertation we explore a low-cost OpenFlow Software Defined Networking solution and compare its performance to that of MPLS.

In summary, we develop methods for designing robust networks and evaluate software defined networking for communication and control in cyber physical systems where the smart grid is the system under consideration.

Table of Contents

Table of Contents	viii
List of Figures	xi
List of Tables	xv
Acknowledgements	xvii
Dedication	xviii
1 Introduction	1
1.1 Communication Infrastructure and Network Technologies for Cyber Physical Systems	1
1.2 Motivation: The Bottom-Up Approach for Resilient Communication Infrastructure	2
1.2.1 The Physical Communication Infrastructure	3
1.2.2 Network Technologies for the Communication Infrastructure	4
1.3 Contributions	5
1.3.1 Organization	6
1.3.2 List Of Symbols	7
2 Improving the Robustness of the Physical Communication Infrastructure	8
2.1 Background and Related Work	10
2.2 Principles of Algebraic Connectivity	11
2.3 The Result of Removing an Edge	13
2.3.1 Upper bound for $\lambda_2(G - e)$	14
2.3.2 Lower bound for $\lambda_2(G - e)$	14

2.4	Network Models	22
2.4.1	Watts-Strogatz Model (WS)	22
2.4.2	Gilbert Stochastic Model (Gi)	23
2.4.3	Barabási-Albert Scale Free Model (BA)	25
2.5	Numerical Analysis for Edge Removal	26
2.6	Comparative Analysis of the Increase in Algebraic Connectivity via Edge Rewiring	27
2.7	Edge Rewiring to Maximally Increase Algebraic Connectivity	30
2.8	Rewiring vs Adding edges to Maximally Increase Algebraic Connectivity	33
2.9	Discussion	33
2.10	Summary	36
3	The Impact of Improving the Robustness of the Physical Communication Infrastructure for Transmission Operations in Smart Grids	38
3.1	Power System Dynamics Model	39
3.2	Communication Network Model For the Transmission Network	42
3.3	Simulation Study	43
3.3.1	Impact to Topological Characteristics of a Network as Algebraic Connectivity is Maximally Increased	44
3.3.2	Impact to Characteristics of Traffic as Algebraic Connectivity is Maximally Increased	46
3.4	Discussion	53
3.5	Summary	53
4	Simulative Comparison of Multiprotocol Label Switching and OpenFlow Network Technologies for Transmission Operations	55
4.1	Simulative Communication Network Model	56
4.1.1	MPLS	57
4.1.2	OpenFlow	57

4.2	Simulation Studies	59
4.2.1	Simulation study #1	60
4.2.2	Simulation study #2	65
4.3	Summary	70
5	Software Defined Networking (SDN) in GENI: Experimental Evaluation of Open-Flow Technology for Smart Grids	71
5.1	Background and Related Work	72
5.2	Power and Smart Grid Model	74
5.2.1	The Electro-Mechanical Model	75
5.2.2	The Smart Grid Prototype	76
5.2.3	OpenFlow Architecture	78
5.3	Experiments	79
5.3.1	Software Defined Networking-Traffic Engineering (SDN-TE)	79
5.3.2	Demand Response	81
5.3.3	Experiment 1: Automatic Fail-over	83
5.3.4	Experiment 2: Congestion	85
5.3.5	Experiment 3: Load Balancing	89
5.3.6	Fast Reroute (SDN-TE Protection)	91
5.4	Discussion	95
5.5	Summary	96
6	Conclusions and Future Research	98
6.0.1	Review	98
6.0.2	Future Work	101
	Bibliography	103

List of Figures

1.1	Phase angle analysis conducted by NERC following the 2003 blackout in the Northeastern states	2
2.1	Construction of Watts-Strogatz model	23
2.2	Node degree distribution for Gilbert Stochastic Model	24
2.3	Node degree distribution for Barabási-Albert Scale Free Model	26
2.4	Decrease in $\lambda_2(G - e)$ for the Watts-Strogatz, Gilbert stochastic, and Barabási-Albert Scale Free networks, respectively	28
2.5	Distributions for the average values of λ_2^i and λ_2^f for Watts-Strogatz (WS), the Barabási–Albert Scale Free (BA), and the Gilbert stochastic (Gi) networks . . .	29
2.6	Distributions for the average values of $\lambda_2^f - \lambda_2^i$ for Watts-Strogatz (WS), the Barabási-Albert Scale Free (BA), and the Gilbert stochastic (Gi) networks	29
2.7	Increase in algebraic connectivity for Watts-Strogatz (WS), the Barabási-Albert Scale Free (BA), and the Gilbert stochastic (Gi) networks as edges are rewired by first inserting an edge then removing another	32
2.8	Increase in algebraic connectivity for Watts-Strogatz (WS), the Barabási-Albert Scale Free (BA), and the Gilbert stochastic (Gi) networks such that $N = 400$. .	33
2.9	Comparison of edge addition to edge rewiring to maximally increase algebraic connectivity in the WS, Gi, and BA networks, respectively.	35
2.10	Comparison of edge addition to edge rewiring to optimize algebraic connectivity in the WS, Gi, and BA networks for $N = 800$	35
3.1	Lattice	39

3.2	Model of the transmission communication network that is based on the IEEE 118 bus test case of the Power System’s Test Case Archive	42
3.3	Number of bytes lost at each interface for all nodes in the <i>LowBW</i> network . . .	47
3.4	Number of bytes lost at each interface for all nodes in the <i>1%Add₁</i> network . . .	47
3.5	Number of bytes lost at each interface for all nodes in the <i>5%Add₁</i> network . . .	48
3.6	Number of bytes lost at each interface for all nodes in the <i>10%Add₁</i> network . . .	48
3.7	Number of bytes lost at each interface for all nodes in the <i>1%Add₂</i> network . . .	49
3.8	Number of bytes lost at each interface for all nodes in the <i>5%Add₂</i> network . . .	49
3.9	Number of source-destination flows traversing the interfaces of each node in the respective inter-substation network	50
3.10	Comparison of throughput and PLR for networks in scenario 1 and networks in scenario 2	52
3.11	Comparison of throughput and RTT for networks in scenario 1 and networks in scenario 2	52
4.1	Communication network for IEEE 118 bus case of the Power System’s Test Case Archive.	60
4.2	Comparison of initial and final voltage profiles for a high bandwidth OpenFlow/MPLS communication network where only protection traffic exists.	62
4.3	Comparison of resulting frequency profiles for the MPLS and OpenFlow networks	63
4.4	Comparison of resulting frequency profiles for the MPLS and OpenFlow networks	63
4.5	Comparison of resulting rate of load adjustment for MPLS and OpenFlow networks	64
4.6	Comparison of throughput (traffic generated and received) for a high bandwidth MPLS/OpenFlow communication network where only protection traffic exists . .	65
4.7	Comparison of resulting frequency profiles for MPLS and OpenFlow communication networks	66
4.8	Comparison of resulting rate of load adjustment when using an MPLS/OpenFlow network	66

4.9	Comparison of resulting frequency profiles for MPLS and OpenFlow communication networks considering the 300 bus system	67
4.10	Comparison of resulting rate of load adjustment when using an MPLS/OpenFlow network	67
4.11	Comparing throughput for a low bandwidth MPLS/OpenFlow communication network where background and protection traffic exists	68
4.12	Comparing throughput for the 300 bus case.	69
5.1	Model of the smart grid	75
5.2	Prototype of the power grid	76
5.3	Integration of the power grid and communication network to realize a prototype of the smart grid	77
5.4	OpenFlow architecture	78
5.5	OpenFlow traffic engineering services on GENI	81
5.6	Comparison of the frequency response for a failure at G3 as an automatic fail-over mechanism reroutes traffic through the backup path to the benchmark experiment where CCA resided at KSU	84
5.7	Throughput through the backbone network as automatic fail-over ensues	85
5.8	Comparison of the the load shed profile as an automatic fail-over mechanism reroutes traffic from the primary to the backup path	85
5.9	Max rate feature for queues on the Pronto 3290 OpenFlow switch	86
5.10	Rate at which packets are received at the Control Center from the GA	87
5.11	Throughput in the primary path as streams are incrementally traversing the network	87
5.12	Comparison of the frequency responses for each trial where x stream/s traversed the network in addition to protection traffic from LA	88
5.13	Rate at which packets are received at the Control Center from the LA	89
5.14	Throughput on both paths as streams are load balanced in the backbone core network	90

- 5.15 Comparison of frequency response for the QoS, load balancing and benchmark experiments 90
- 5.16 Comparison of the load shed profile for the QoS, load balancing and benchmark experiment 91
- 5.17 MPLS network configuration 92
- 5.18 OpenFlow network configuration 93
- 5.19 Comparison of OpenFlow and MPLS FRR mechanisms as ICMP ping packets are transmitted 94
- 5.20 Comparison of OpenFlow and MPLS FRR mechanisms as Iperf TCP packets are transmitted 95

List of Tables

1.1	Definitions of the most common variables used throughout this dissertation. “Vertices” and “edges” are used within the graph theory domain, and “nodes” and “links” are used when referring to a physical network within the communication networking domain. In any case, a “vertex” is synonymous with a “node” and an “edge” is synonymous with a “link.”	7
2.1	Elements of the table correspond to the number of edges for the specified network with size N	30
3.1	Network parameters for the Case Study	43
3.2	Background traffic parameters for the transmission network	43
3.3	Impact on topological characteristics of a network as algebraic connectivity is maximally increased	45
3.4	Average number of bytes lost at each interfaced, totaled over all interfaces for scenarios 1 and 2	46
3.5	Impact on network characteristics as algebraic connectivity is increased in scenario 1	51
4.1	Network parameters for simulation study 1 (SS1), and 2	61

Acknowledgments

The completion of this dissertation marks the close of yet another chapter in my pursuit of excellence. First of all, I am most grateful for the wisdom bestowed by Jesus Christ and His daily guidance that allowed me to navigate the trials of these past four years. Secondly, I would like to thank Dr. Soldan for providing the avenue to pursue my education at K-State. Thirdly, this work would not be a reality without the support and encouragement of my advisors, Dr. Scoglio and Dr. Gruenbacher, under whose supervision I began and completed this dissertation. Their engaging personalities constantly motivated me to achieve excellence by ensuring all fundamental concepts were thoroughly understood. Furthermore, their assistance in funding and research resulted in two consecutive internships at Raytheon BBN Technologies which were vital to my research vision and direction. I would also like to thank my supervisors at Raytheon BBN, Heidi Picher Dempsey and Harry Mussman, for demonstrating confidence in my capabilities and for broadening my perspective in the field of networking. I cannot forget the constant assistance from network experts at BBN, including Niky Riga, Tim Upthegrove, Josh Smith, and Chaos Golubitsky, for their prompt assistance in troubleshooting issues related to the GENI testbed. I would also like to recognize the support of our ECE departmental IT staff: Samuel Hays, Nathanael Brooks, and Steven Booth.

I would like to thank members from my church family, including Madelyn and John Clarke who provided a home away from home. John constantly reminded me that I was a “Genius” and even though I thought of myself as an ordinary guy, this positive thought constantly propelled me to work harder toward becoming a genius. I still have a long way to go though (smiling). Sharing my love for music, Madelyn and John also provided a music room where the youth from our church could meet and Jam: Friday nights were our praise and worship jam sessions at the Upper Room and the Broken Arrow Ranch.

From the research domain, I would like to acknowledge support from the Department of Energy (DOE), Burns & McDonnell, the Electrical Power Affiliates Program (EPAP), Kansas Re-

search and Education Network (KanREN) and FishNet Securities. Furthermore, I would like to thank Netta Khodra for editing all my publications. Her constructive, grammatical criticisms expedited delivery and acceptance of these publications. I would like to thank my research group, Sunflower Networking Research, for their friendship and sacrifices. In particular, I'd like to recognize Phillip Schumm for insights which played a critical role in my work.

I cannot end without thanking my family: Victoria, Andrew, Curby Dwaine, and Tayilah Sydney, whose constant thoughtfulness and love has provided this thirst for knowledge. I am also appreciative of my older brother's example in being my role model, Dr. Curby Dwaine Sydney, who never rests until his objective has been achieved. As a specialist in neurosurgery, I am always inspired by his exceptional work ethic and seek to emulate these character traits. Furthermore, I would like to thank my church family back home in St Lucia, especially the Khodras, for their constant prayers. Finally, I want to thank Netta Khodra for her support and encouragement throughout these years.

Dedication

To my loving sister, Taliyah Shekiel Vican Sydney whom I have not spent much time with as I have been away pursuing this degree. Post graduation, I intend to be in her life like the sun in the sky.

To my parents Victoria and Andrew Sydney for taking an active interest in my education, moral, and spiritual development.

To my brother Curby Dwaine Sydney for being my role model.

To Netta Khodra for her consistent support.

To my community at La Croix Maingot, St Lucia. May this work demonstrate that greatness lies not in ordinary things, but in ordinary things done extra-ordinarily well.

Chapter 1

Introduction

1.1 Communication Infrastructure and Network Technologies for Cyber Physical Systems

Why study next-generation communication architecture for cyber physical systems, particularly, the Electric Power Grid? Primarily the answer is because we have experienced moderate-scale power system failures within the US and abroad, and thus large-scale failures are inevitable as the load on the aging infrastructure increases. One classic and riveting example of such failure stems from a series of cascading failures in 2003 that resulted in a blackout in the Northeastern states [1]. Figure 1.1 was extracted from the post-event analysis conducted by the North American Electric Reliability Corporation (NERC) in which over 50 million people and over 400 generators were affected [2]. In particular, between the period of 15:50 and 16:10, the angular separation leaped from 25 to 115 degrees, 90 degrees from the normal operating condition. Coincidentally, a similar phenomenon occurred the very same year in Italy, leaving 56 million residents without power for 9 hours [3].

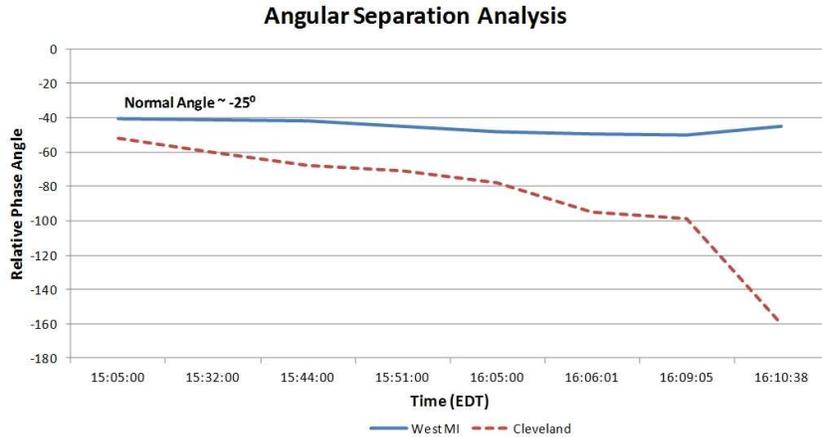


Figure 1.1: Phase angle analysis conducted by NERC following the 2003 blackout in the North-eastern states

One common factor during blackouts is the lack of situational awareness [4, 5, 6, 7]. In the case of the USA/Canada 2003 blackout, the initial stages began well over an hour before the cascading failures ensued. Nevertheless, with low visibility of the entire power system, both human and computer reactions were too slow to mitigate a blackout. Several projects, including GridStat, are dedicated to increasing grid awareness by augmenting current technologies, such as synchphasers, with GridStat middle-ware [8, 9]. However, all of these technologies depend on the communication infrastructure meeting current demands of continuous availability, reliability, and efficiency. Consequently, the smart grid communication infrastructure must be adapted and perhaps re-engineered to meet these disparate demands.

1.2 Motivation: The Bottom-Up Approach for Resilient Communication Infrastructure

Though problems such as congestion in communication networks are addressed by protocols, middle-ware, and software mechanisms, one should not underestimate the significance of the physical infrastructure. Furthermore, a network whose physical topology is designed to address problems, such as congestion, realizes an even higher level of performance. For this reason, we

revisit the physical topology to determine methods that yield robust topologies before evaluating the various technologies for smart grid communication. Subsequently, we compare performance of the current MPLS networking technology to the proposed OpenFlow Software Defined Networking solution.

1.2.1 The Physical Communication Infrastructure

To date, the topology of the communication infrastructure tends to be identical to that of the power grid infrastructure, but a topology that seems ideal for the power network, may not necessarily be ideal for the communication network. Therefore, one objective of this dissertation is to determine methods that yields communication network topologies with high performance characteristics. To this end, we consider key principles of graph theory to obtain robust communication networks.

Robustness in complex networks is an ongoing research effort that seeks to improve the connectivity of networks against attacks and failures. Among other measures, algebraic connectivity, a metric from the domain of spectral analysis in graph theory, has been used to characterize processes such as damped oscillation of liquids in connected pipes. Similar characterizations include the number of edges necessary to disconnect a network; namely, the larger the algebraic connectivity, the larger the number of edges required to disconnect a network and hence, the more robust a network. In this dissertation, we answer the question, “Which edge can we rewire to have the largest increase in algebraic connectivity?” Furthermore, we extend the rewiring of a single edge to rewiring multiple edges in order to realize the maximal increase in algebraic connectivity. The answer to the previous question can provide insights for decision makers within domains such as communication and transportation networks, who seek an efficient solution to optimizing connectivity and thus increasing the robustness of their networks. Most importantly, our analytical and numerical results not only provide insights as to the number of edges to rewire, but also the location in the network where these edges would effectuate the maximal increase in algebraic connectivity and therefore enable a maximal increase in robustness.

Our analytical and numerical results are based on theoretical principles and models. For this

reason, the question still remains, “What is the impact on the characteristics of real-world networks when algebraic connectivity is maximized?” In response to this question, we conduct an analysis on the impact of algebraic connectivity maximization on characteristics of the network topology. Subsequently, we use a hybrid simulator that integrates the power system and communication network and conduct an analysis on the impact of algebraic connectivity maximization on the reduction of traffic congestion.

1.2.2 Network Technologies for the Communication Infrastructure

Currently, utilities are gravitating towards technologies such as MPLS because of proven reliability over the years and mechanism provision for efficient overlay technologies. In particular, MPLS satisfies NERC’s Critical Infrastructure Protection standard (CIP-002) which stipulates that traffic to critical assets (assets that, if targeted, can affect the bulk power system) should be sent over Layer 2, as defined by the Open System Interconnection (OSI) model [5, 10, 11, 12]. Additionally, MPLS provides traffic engineering and virtual private network (VPNs) services. These services rely on multiple protocols, such as Open Shortest Path First (OSPF) and Resource Reservation protocol (RSVP). In addition, all routers must enable new protocols to support any new network services. Extensive tests can run from three to 10 years and must be conducted to deploy these new services in order to minimize service interruptions [13]. In any case, with technologies such as MPLS, innovation is restricted to the features enclosed “...in the box.”

Conversely, OpenFlow’s flexibility and programmability realizes a control plane that provides similar functionalities to MPLS. With increasing support from network providers such as Microsoft, Google, Amazon and equipment vendors such as NEC, Juniper, Cisco, and Brocade, OpenFlow’s modularity implies that changes to network services require a simple change in the OpenFlow controller deployed on the network operating system [13, 14, 15]. Furthermore, with OpenFlow, new services are not tied to extensions of existing protocols, unlike MPLS in which new services such as RSVP-TE (RSVP-Traffic Engineering) are tied to RSVP. In addition to these advantages, OpenFlow’s ability to isolate network traffic ensures that failure of an experimental

protocol, service, or application does not affect other experiments or hinder production traffic. In the same way, different classes of traffic in the smart grid can be isolated for Quality of Service (QoS) guarantees. For these reasons, OpenFlow may provide a more capable backbone communication technology that is overall less expensive than MPLS.

For our research, we first deploy a simulative prototype of the smart grid to demonstrate that OpenFlow performs as well as MPLS and may, therefore, be considered an alternative to MPLS for smart grid applications. Though previous research demonstrates that OpenFlow can provide similar services as MPLS using Open VSwitch software switches, the current OpenFlow hardware does not readily support MPLS [13]. Therefore, we deploy a real-world prototype of the transmission component of the smart grid to demonstrate the flexibility and programmability of OpenFlow in providing services similar to MPLS. It is worth noting that this work is a proof of concept; therefore, software verification and validation are outside the scope of this dissertation.

1.3 Contributions

In keeping with our bottom-up approach for designing a resilient smart grid communication infrastructure, we contribute the following:

- Two corollaries to develop framework for constructing the upper and lower bounds for algebraic connectivity when an edge is removed
- A method to select the edge that, when removed, decreases algebraic connectivity the least
- An algorithm that removes edges to numerically validate our analytical results for the upper and lower bounds. Additionally, we present a second algorithm to rewire edges and a third algorithm to add edges to maximally increase algebraic connectivity. All algorithms have a running time $O(|V|^2)$
- The comparison of three network models to determine which one realizes the highest increase in algebraic connectivity when a small percentage of the edges are rewired while keeping the number of nodes and edges constant

- A comparison of the performance of MPLS and OpenFlow networks in the context of a hypothetical smart grid application.
- A study to illustrate that maximizing algebraic connectivity results in a more homogeneous network topology
- A study to show that maximizing algebraic connectivity reduces the level of traffic congestion in a network
- An OpenFlow controller that implements an automatic fail-over mechanism in addition to a Quality of Service (QoS) queuing mechanism. This controller also highlights Software Defined Networking-Tunnel Engineering (SDN-TE) features such as auto-route, auto-bandwidth, load balancing, priorities, flow preemption, and fast reroute.
- A Demand Response (DR) smart grid application that transmits traffic created by cyber physical systems

1.3.1 Organization

The following outlines the structure of this dissertation. Chapters 2 and 3 address physical communication infrastructure. In particular, Chapter 2 delves into the analytical and numerical principles used to determine methods for increasing the algebraic connectivity of the physical network. In Chapter 3, we utilize the methods obtained in Chapter 2 to study the impact of increasing algebraic connectivity on both the topology and the traffic characteristics of real-world smart grid models. Chapters 4 and 5 address networking technologies used for the communication infrastructure of the smart grid. More specifically, Chapter 4 is based on a hybrid simulator that compares the performance of MPLS to OpenFlow for transmission operation of the smart grid. In Chapter 5, we deploy a real-world prototype of a smart grid to demonstrate the capability of OpenFlow to provide similar services as MPLS using power components from K-State and networking components of the Global Environment for Network Innovations (GENI) testbed. Finally, Chapter 6 concludes this dissertation by discussing the applicability and benefits of this work in evaluating

software defined networks for communication and control of cyber physical systems. We also provide guidance as to possible directions for this work.

1.3.2 List Of Symbols

Table 1.1 lists the most common variables used throughout this dissertation.

Variables	Definitions
V	set of vertices
v	vertex v
$deg(v)$	degree of vertex v
$\overline{deg(v)}$	average vertex degree of vertex v
E	set of edges
e	edge e
λ	eigenvalue
u, w, z	vectors
N	total number of vertices
G	graph
A	adjacency matrix
L	laplacian matrix
D	diameter
R	radius
CPL	characteristic path length
C_{coe}	clustering coefficient
H	heterogeneity

Table 1.1: *Definitions of the most common variables used throughout this dissertation. “Vertices” and “edges” are used within the graph theory domain, and “nodes” and “links” are used when referring to a physical network within the communication networking domain. In any case, a “vertex” is synonymous with a “node” and an “edge” is synonymous with a “link.”*

Chapter 2

Improving the Robustness of the Physical Communication Infrastructure

To improve the robustness of the physical communication infrastructure, we explore algebraic connectivity: a spectral measure to determine the robustness of a graph. As a topological measure, we recognize the limitations of algebraic connectivity when used as the determining factor to increase the robustness of a real-world network [16, 17, 18, 19]. For such networking domains, other measures particular to the behavior of the considered network can be used in addition to algebraic connectivity in order to provide a comprehensive solution to increase the robustness of a network.

In this chapter, we endeavor to answer the question of where an edge should be rewired to increase algebraic connectivity the most. Our approach is based on studies conducted to determine where an edge should be added to increase algebraic connectivity the most [20, 21]. Given a network $G(V, E)$ such that $|V|$ is the number of vertices and $|E|$ is the number of edges, the number of possibilities to rewire an edge is given by $\binom{|V|}{2} - |E|$. For complex networks, comparing each edge to find the optimal one that maximizes algebraic connectivity is infeasible. Furthermore, as a complimentary problem, it has been proven that maximum algebraic connectivity augmentation is NP-Hard [22]. For this reason, we propose a strategy that rewires edges to maximally increase the algebraic connectivity of a network.

In our approach, we consider the rewiring of an edge as a two-step process in which we either

insert an edge and then remove an edge, or we remove an edge and then insert an edge. Hence, our original question of “Where should an edge be rewired to increase algebraic connectivity the most?” is subdivided into two parts:

1. “Where should an edge be removed to decrease algebraic connectivity the least?”
2. “Where should an edge be added to increase algebraic connectivity the most?”

The latter question has been addressed [20, 21]. Therefore, this chapter focuses on the first question and contributes the following:

- Two corollaries to develop framework for constructing the upper and lower bounds for algebraic connectivity when an edge is removed
- A method to select the edge that, when removed, decreases algebraic connectivity the least
- An algorithm that removes edges to numerically validate our analytical results for the upper and lower bounds. Additionally, we present a second algorithm to rewire edges and a third algorithm to add edges in order to maximally increase algebraic connectivity. All algorithms have a running time $O(|V|^2)$.
- The comparison of three network models to determine which one realizes the highest increase in algebraic connectivity when a small percentage of the edges are rewired while maintaining a constant number of nodes and edges.

The structure of this chapter is outlined as follows: Section 2.1 builds on the Introduction by providing the necessary background and state-of-the-art for algebraic connectivity. Section 2.2 reviews theorems and definitions, and introduces two corollaries to two of the theorems presented. Section 2.3 presents the lower and upper bounds for algebraic connectivity when an edge is removed, and in Section 2.4, we review the three network models used in our analysis: Watts-Strogatz model, Gilbert’s stochastic model, and Barabási-Albert Scale Free Model. Section 2.5 describes an algorithm for edge removal, and also we provide the numerical analysis for edge

removal for the three classes of networks. In Section 2.6, we compare graphs from three different models to determine which model realizes the greatest increase in algebraic connectivity through rewiring. Section 2.7 presents a second algorithm and the corresponding implementation to rewire edges to maximally increase algebraic connectivity. In Section 2.8, we present a third algorithm to add edges to increase algebraic connectivity. A comparison is then drawn based on results of adding edges to that of rewiring edges to maximally increase algebraic connectivity. Section 2.9 presents a discussion on the applicability of this work in the real world, and, finally, Section 2.10 discusses the benefits and shortcomings of the rewiring approach.

2.1 Background and Related Work

The classical approach for determining robustness of networks entails the use of basic graph theory concepts. For instance, the connectivity of a graph is an important, and probably the earliest, measure of robustness of a network [23]. Vertex (edge) connectivity, defined as the size of the smallest vertex (edge) cut, in a certain sense determines the robustness of a graph to deletion of vertices (edges). However, the vertex or edge connectivity only partly reflects the ability of graphs to retain certain degrees of connectedness after deletion. Other improved measures were introduced and studied, including super connectivity [24], conditional connectivity [25], restricted connectivity [26], fault diameter [27], toughness [28], scattering number [29], tenacity [30], expansion parameter [31], and isoperimetric number [32]. In contrast to vertex (edge) connectivity, these new measures consider both the cost to damage a network and how extensively the network is damaged.

Subsequent measures consider the size of the largest connected component as vertices are attacked [33]. Furthermore, percolation models were used to assess the damage incurred by random graphs [34]. More recent efforts present a topological analysis of robustness in networks such as the power grid [35]. Other metrics in networking literature include the average node degree [36], betweenness [37], heterogeneity [38], and characteristic path length [39].

The measures reviewed thus far, consider the network structure to assess robustness. However,

recent efforts have incorporated the behavior of the network to assess robustness, maximizing flows in the network while imposing constraints on routers and links [16, 18, 19].

From spectral analysis, experimentalists have generally utilized the second smallest laplacian eigenvalue to guarantee connectivity of a graph; if this value is 0, a graph is disconnected [40]. Furthermore, several relationships, such as network diameter, have been established between algebraic connectivity and graph theoretical measures and are relevant to domains like the Internet in order to understand the implications of protocols, such as spanning tree [41, 42]. In the area of robustness, the second smallest eigenvalue has also been considered as a measure of the difficulty of breaking the network into components [43]. This eigenvalue, called the algebraic connectivity of a graph, has been extracted from the admittance spectrum and used to characterize both the flows through communicating pipes and also the permeability of graphs [20]. Furthermore, the concept of algebraic connectivity was used to determine where to add an edge in order to maximally increase algebraic connectivity. The results from this work were implemented numerically [21]. Finally, the bounds for algebraic connectivity were derived by applying Rayleigh's theorem which, as Section 2.2 explains, is also used to derive the lower bound when an edge is removed[44].

2.2 Principles of Algebraic Connectivity

Throughout this chapter, $G = (V, E)$ is an undirected, connected graph with vertex set $V = 1, \dots, N$ and edge set E , such that $N = |V|$ is the number of vertices. u, w, z are vectors, λ is an eigenvalue, and $deg(v)$ is the vertex degree of vertex $v \in V$.

Definition 1. *Given a graph G , the Laplacian $L(G)$ of G is an $N \times N$ matrix L defined by*

$$L_{ij} = \begin{cases} deg(i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } (i, j) \in E \\ 0 & \text{if } i \neq j \text{ and } (i, j) \notin E \end{cases}$$

$L(G)$ is a symmetric positive semidefinite matrix with all real and non-negative eigenvalues. The set of eigenvalues denoted by $\lambda_1(G) \leq \lambda_2(G) \leq \dots \leq \lambda_N(G)$, is the Laplacian spectrum of

graph G .

Definition 2. *The algebraic connectivity of a graph G is the second-smallest eigenvalue of $L(G)$: $\lambda_2(G)$*

Theorem 1. *Let G be a graph with N vertices. Let $G + e$ be the augmented graph obtained by adding edge e between two vertices in G . Then the eigenvalues of G and $G + e$ are intertwined as follows [45]:*

$$0 = \lambda_1(G) \leq \lambda_1(G + e) \leq \lambda_2(G) \leq \lambda_2(G + e) \leq \dots \leq \lambda_N(G) \leq \lambda_N(G + e).$$

If $\lambda_2(G)$ is a multiple eigenvalue such that $\lambda_2(G) = \lambda_2(G + e)$, the result of adding an edge does not improve the algebraic connectivity. Given that the trace(L) = $\sum_{i=1}^N \lambda_i(G) = 2|E|$, it follows that

$$\sum_{i=1}^N (\lambda_i(G + e) - \lambda_i(G)) = 2 \tag{2.1}$$

which implies that $0 \leq \lambda_2(G + e) - \lambda_2(G) \leq 2$. Additionally, we deduce that given a graph with N vertices, the magnitude of λ_i for $i \in N$ tends to increase as $|E|$ increases.

Corollary 1. *Let G be a graph with N vertices. Let $G - e$ be the augmented graph obtained by removing an edge e between two vertices in G such that the removal of an edge does not disconnect the graph. Then the eigenvalues of G and $G - e$ are intertwined as follows:*

$$0 = \lambda_1(G - e) \leq \lambda_1(G) \leq \lambda_2(G - e) \leq \lambda_2(G) \leq \dots \leq \lambda_N(G - e) \leq \lambda_N(G).$$

We can also deduce that:

$$\sum_{i=1}^N (\lambda_i(G) - \lambda_i(G - e)) = 2 \tag{2.2}$$

This implies that $0 \leq \lambda_2(G) - \lambda_2(G - e) \leq 2$ and that given a graph with N vertices, the magnitude of λ_i for $i \in N$ tends to increase as $|E|$ increases.

Theorem 2 provides the condition under which algebraic connectivity increases by 2.

Theorem 2. *Let G be a connected graph with N vertices and let i and j be two non-adjacent vertices in G . The largest possible increase in algebraic connectivity occurs if and only if $G = K_N \setminus \{i, j\}$: the complete graph with one edge removed [46].*

Corollary 2. *Let G be a connected graph with N vertices and let i and j be two non-adjacent vertices in G . The largest possible decrease in algebraic connectivity occurs if and only if $G = K_N$: the complete graph.*

Theorem 3. *Let G be a simple connected graph with $N > 2$. If G has a pendant vertex (i.e. a vertex with degree 1), $\lambda_2 \leq 1$. Moreover, $\lambda_2 < 1$ if the pendant vertex is not adjacent to the highest degree vertex [47].*

Complex networks typically contain pendant vertices and for this reason $\lambda_2(G) < 1$. This implies that $\lambda_2(G - e) < 1$.

2.3 The Result of Removing an Edge

The removal of edge $v_i v_j$ from G for $i, j \in V$ can be achieved using a positive semidefinite matrix B . An example of B such that $i = 1$ and $j = 2$ is shown below:

$$\begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Thus, for the spectrum $\lambda_1(G - e), \dots, \lambda_N(G - e)$ of $L - B$, we have

$$0 = \lambda_1(G - e) = \lambda_1(G) \leq \lambda_2(G - e) \leq \lambda_2(G) \leq \dots \leq \lambda_N(G - e) \leq \lambda_N(G).$$

2.3.1 Upper bound for $\lambda_2(G - e)$

Given that $v_i v_j$ are the vertices from which an edge is removed, let z be a vector with $+1$ for the i th component, -1 for the j th component and 0 otherwise. Additionally, let $u_i^{(2)}$ represent the i th element of the eigenvector that corresponds to λ_2 : the second smallest eigenvalue. It follows that our matrix $B = zz^T$. Also, let $\alpha := |(z, u^{(2)})| = \left| u_i^{(2)}(G) - u_j^{(2)}(G) \right|$, such that $(i, j) \in E$: the set of edges of G . For a vector $w \perp u^{(1)}(G - e)$, and assuming $u^{(2)}(G) = w$ the Rayleigh quotient has the following property:

$$\begin{aligned} R(u^{(2)}) &= u^{(2)T}(L - B)u^{(2)} \\ &= \lambda_2 - u^{(2)T}zz^T u^{(2)} \\ &= \lambda_2 - \alpha^2 \end{aligned} \tag{2.3}$$

Therefore,

$$\lambda_2(G - e) \leq \lambda_2(G) - \alpha^2$$

From the upper bound for $\lambda_2(G - e)$, we deduce that the lower α is (that is, the smaller the difference between elements on the eigenvector corresponding to the second smallest eigenvalue), the higher the upper bound.

2.3.2 Lower bound for $\lambda_2(G - e)$

To obtain the lower bound, we use the technique of intermediate value problems [48]. Our new laplacian $L' = L - zz^T$. To make zz^T positive definite, we replace it by $C := -zz^T - \epsilon I$. If we let $k = 2$, $p^{(r)} := C^{-1}u^{(r)}$, such that $r = 1, \dots, k$, we get the matrix $(\gamma_{r,s})_{r,s=1,2} := (\langle p^{(r)}, Cp^{(s)} \rangle)_{r,s=1,2}^{-1}$

$$\begin{pmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{pmatrix}$$

In particular, each element of the matrix can be obtained by first expanding $\gamma_{r,s}$ as follows:

$$\begin{aligned}
(\gamma_{r,s}) &= (p^{(r)}, Cp^{(s)})^{-1} \\
&= [p^{(r)}, (-zz^T - \epsilon I)p^{(s)}]^{-1} \\
&= (p^{(r)}, -zz^T p^{(s)} - \epsilon p^{(s)})^{-1} \\
&= \left[(p^{(r)})^T (-zz^T p^{(s)} - \epsilon p^{(s)}) \right]^{-1} \\
&= \left[(-p^{(r)})^T zz^T p^{(s)} - \epsilon (p^{(r)})^T p^{(s)} \right]^{-1}
\end{aligned} \tag{2.4}$$

Secondly, given the nonsingular $N \times N$ matrix A and vector z , we use the following formula by Sherman-Morrison [49]:

$$(zz^T + A)^{-1} = A^{-1} - \frac{A^{-1}zz^T A^{-1}}{1 + z^T A^{-1}z} \tag{2.5}$$

to obtain the inverse of C as follows:

$$C^{-1} = \frac{1}{\epsilon} I - \frac{zz^T}{\epsilon^2 + \epsilon |z|^2} \tag{2.6}$$

Therefore, if $r = s = 1$, $(\gamma_{11})^{-1}$ can be computed as follows:

$$(\gamma_{11})^{-1} = - [C^{-1}u^{(1)}]^T zz^T [C^{-1}u^{(1)}] - \epsilon [C^{-1}u^{(1)}]^T [C^{-1}u^{(1)}] \tag{2.7}$$

We can reduce equation 2.7 by considering each block as follows:

$$\begin{aligned}
[C^{-1}u^{(1)}]^T &= \left[\frac{u^{(1)}}{\epsilon} - \frac{zz^T u^{(1)}}{\epsilon^2 + \epsilon |z|^2} \right] \\
&= \frac{(u^{(1)})^T}{\epsilon} - \frac{(u^{(1)})^T zz^T}{\epsilon^2 + \epsilon |z|^2}
\end{aligned} \tag{2.8}$$

$$\begin{aligned}
[C^{-1}u^{(1)}]^T zz^T &= \frac{(u^{(1)})^T zz^T}{\epsilon} - \frac{(u^{(1)})^T zz^T zz^T}{\epsilon^2 + \epsilon |z|^2} \\
&= \frac{(u^{(1)})^T zz^T}{\epsilon} - \frac{|z|^2 (u^{(1)})^T zz^T}{\epsilon^2 + \epsilon |z|^2} \\
&= (u^{(1)})^T zz^T \left[\frac{1}{\epsilon} - \frac{|z|^2}{\epsilon^2 + \epsilon |z|^2} \right] \tag{2.9}
\end{aligned}$$

$$[C^{-1}u^{(1)}]^T zz^T [C^{-1}u^{(1)}] = (u^{(1)})^T zz^T \left[\frac{1}{\epsilon} - \frac{|z|^2}{\epsilon^2 + \epsilon |z|^2} \right] \left[\frac{u^{(1)}}{\epsilon} - \frac{zz^T u^{(1)}}{\epsilon^2 + \epsilon |z|^2} \right] \tag{2.10}$$

If we factor the constant term $\left[\frac{1}{\epsilon} - \frac{|z|^2}{\epsilon^2 + \epsilon |z|^2} \right]$, we obtain the following:

$$\begin{aligned}
& \left[\frac{1}{\epsilon} - \frac{|z|^2}{\epsilon^2 + \epsilon |z|^2} \right] (u^{(1)})^T z z^T \left[\frac{u^{(1)}}{\epsilon} - \frac{z z^T u^{(1)}}{\epsilon^2 + \epsilon |z|^2} \right] = \left[\frac{1}{\epsilon} - \frac{|z|^2}{\epsilon^2 + \epsilon |z|^2} \right] \\
& \left[\frac{(u^{(1)})^T z z^T u^1}{\epsilon} - \frac{(u^{(1)})^T z z^T z z^T u^{(1)}}{\epsilon^2 + \epsilon |z|^2} \right] \tag{2.11} \\
& = \left[\frac{1}{\epsilon} - \frac{|z|^2}{\epsilon^2 + \epsilon |z|^2} \right] \left[\frac{(u^{(1)})^T z z^T u^1}{\epsilon} - \frac{|z|^2 (u^{(1)})^T z z^T u^{(1)}}{\epsilon^2 + \epsilon |z|^2} \right] \\
& = \left[\frac{1}{\epsilon} - \frac{|z|^2}{\epsilon^2 + \epsilon |z|^2} \right] \left[\frac{(\epsilon^2 + \epsilon |z|^2) (u^1)^T z z^T u^{(1)} - \epsilon |z|^2 (u^1)^T z z^T u^1}{\epsilon (\epsilon^2 + \epsilon |z|^2)} \right] \\
& = \left[\frac{1}{\epsilon} - \frac{|z|^2}{\epsilon^2 + \epsilon |z|^2} \right] \left[\frac{(u^1)^T z z^T u^1 [\epsilon^2 + \epsilon |z|^2 - \epsilon |z|^2]}{\epsilon^2 (\epsilon + |z|^2)} \right] \\
& = \left[\frac{1}{\epsilon} - \frac{|z|^2}{\epsilon^2 + \epsilon |z|^2} \right] \left[\frac{\epsilon^2 (u^{(1)})^T z z^T u^{(1)}}{\epsilon^2 (\epsilon + |z|^2)} \right] \\
& = \left[\frac{1}{\epsilon} - \frac{|z|^2}{\epsilon^2 + \epsilon |z|^2} \right] \left[\frac{(u^{(1)})^T z z^T u^{(1)}}{(\epsilon + |z|^2)} \right] \\
& = \left[\frac{\epsilon^2 + \epsilon |z|^2 - \epsilon |z|^2}{\epsilon (\epsilon^2 + \epsilon |z|^2)} \right] \left[\frac{(u^{(1)})^T z z^T u^{(1)}}{(\epsilon + |z|^2)} \right] \\
& = \left[\frac{1}{\epsilon + |z|^2} \right] \left[\frac{(u^{(1)}) z z^T u^{(1)}}{\epsilon + |z|^2} \right] \\
& = \frac{(u^{(1)})^T z z^T u^1}{(\epsilon + |z|^2)^2} \tag{2.12}
\end{aligned}$$

Therefore,

$$\begin{aligned}
(\gamma_{11})^{-1} &= - [C^{-1}u^{(1)}]^T z z^T [C^{-1}u^{(1)}] - \\
&\quad \epsilon [C^{-1}u^{(1)}]^T [C^{-1}u^{(1)}] \tag{2.13}
\end{aligned}$$

$$\begin{aligned}
&= - \left[\frac{(u^{(1)})^T z z^T u^{(1)}}{(\epsilon + |z|^2)^2} \right] - \epsilon \left| \frac{u^{(1)}}{\epsilon} - \frac{z z^T u^{(1)}}{\epsilon^2 + \epsilon |z|^2} \right|^2 \\
&= - \left[\frac{(u^{(1)})^T z z^T u^{(1)}}{(\epsilon + |z|^2)^2} \right] - \frac{|\epsilon u^{(1)} + |v|^2 u^{(1)} - z z^T u^{(1)}|^2}{\epsilon (\epsilon + |z|^2)^2} \\
&= \frac{-\epsilon (u^{(1)})^T z z^T u^{(1)} - |\epsilon u^{(1)} + |v|^2 u^{(1)} - z z^T u^{(1)}|^2}{\epsilon (\epsilon + |z|^2)^2} \tag{2.14}
\end{aligned}$$

Since $(u^{(1)})^T z = v^T u^{(1)}$,

$$(\gamma_{11})^{-1} = \frac{-\epsilon (z^T u^{(1)})^2 - |\epsilon u^{(1)} + |v|^2 u^{(1)} - z z^T u^{(1)}|^2}{\epsilon (\epsilon + |z|^2)^2} \tag{2.15}$$

Also, since $|\epsilon u^{(1)} + |v|^2 u^{(1)} - z z^T u^{(1)}|^2$ can be expressed as the difference of vectors such that $(a - b)^2 = a^2 - b^2 + 2ab = a^2 + b^2 - 2ab$, we have

$$(\gamma_{11})^{-1} = \frac{-\epsilon (z^T u^{(1)})^2 - |u^{(1)} (\epsilon + |z|^2) - z (z^T u^{(1)})|^2}{\epsilon (\epsilon + |z|^2)^2} \tag{2.16}$$

Since $(\epsilon + |z|^2)$ and $(z^T u^{(1)})$ are scalars and we observe that our vectors $u^{(1)}$ and z can only be multiplied if either is transposed (i.e. $z^T u^{(1)}$ or $(u^{(1)})^T z$), our expression can be reduced as follows:

$$\begin{aligned}
(\gamma_{11})^{-1} &= \frac{-\epsilon (z^T u^{(1)})^2 - \left[|u^{(1)} (\epsilon + |z|^2)|^2 + |z (z^T u^{(1)})|^2 - 2 (\epsilon + |z|^2) (z^T u^{(1)})^2 \right]}{\epsilon (\epsilon + |z|^2)^2} \\
&= \frac{-\epsilon (z^T u^{(1)})^2 - \left[|\epsilon + |z|^2|^2 + |z^T u^{(1)}|^2 |z|^2 - 2 (\epsilon + |z|^2) (z^T u^{(1)})^2 \right]}{\epsilon (\epsilon + |z|^2)^2} \tag{2.17}
\end{aligned}$$

Since $u^{(1)}$ is constant, $z^T u^{(1)} = 0$. Therefore,

$$-\frac{|\epsilon + |z|^2|^2}{\epsilon(\epsilon + |z|^2)^2} = -\frac{1}{\epsilon} \quad (2.18)$$

From this we obtain $\gamma_{11} = -\epsilon$. Using our previous formulations for γ_{11} , if $r = s = 2$, we compute γ_{22} as follows:

$$\begin{aligned} (\gamma_{22})^{-1} &= \frac{-\epsilon(z^T u^{(2)})^2 - |\epsilon + |z|^2|^2}{\epsilon(\epsilon + |z|^2)^2} - \\ &\quad \frac{|z^T u^{(2)}|^2 |z|^2 - 2(\epsilon + |z|^2)(z^T u^{(2)})^2}{\epsilon(\epsilon + |z|^2)^2} \\ &= \frac{(z^T u^{(2)})^2 [\epsilon + |z|^2] - |\epsilon + |z|^2|^2}{\epsilon(\epsilon + |z|^2)^2} \end{aligned} \quad (2.19)$$

Let $\alpha = |z^T u^{(2)}| = |u_i^{(2)} - u_j^{(2)}|$: the difference between the i th and j th elements of $u^{(2)}$, the vector corresponding to the second smallest eigenvalue. Since $|z|^2 = 2$, it follows that:

$$\begin{aligned} (\gamma_{22})^{-1} &= \frac{\alpha^2 [\epsilon + |z|^2] - |\epsilon + |z|^2|^2}{\epsilon(\epsilon + |z|^2)^2} \\ &= \frac{\alpha^2}{\epsilon^2 + \epsilon|z|^2} - \frac{1}{\epsilon} \\ &= \frac{\alpha^2}{\epsilon^2 + 2\epsilon} - \frac{1}{\epsilon} \\ &= \begin{bmatrix} 1 \\ \frac{\alpha^2}{\epsilon^2 + 2\epsilon} - \frac{1}{\epsilon} \end{bmatrix} \begin{bmatrix} \epsilon \\ \epsilon \end{bmatrix} \\ \gamma_{22} &= \frac{\epsilon}{\frac{\alpha^2}{\epsilon + 2} - 1} \end{aligned} \quad (2.20)$$

For $r \neq s$, $\gamma_{rs} = 0$. Therefore, the matrix γ_{rs} is constructed as follows:

$$\begin{pmatrix} -\epsilon & 0 \\ 0 & \frac{\epsilon}{\frac{\alpha^2}{\epsilon + 2} - 1} \end{pmatrix}$$

The intermediate eigenvalue problem corresponding to the second Rayleigh quotient becomes:

$$Lu + \langle u, u^{(1)} \rangle \gamma_{11} u^{(1)} + \langle u, u^{(2)} \rangle \gamma_{22} u^{(2)} = \tau u \quad (2.21)$$

We then use a matrix S to extract the spectrum of L' as follows:

$$\begin{pmatrix} \tau_1 & 0 & 0 & \cdots & 0 \\ 0 & \tau_2 & 0 & \cdots & 0 \\ 0 & 0 & \tau_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \tau_N \end{pmatrix} = S^{-1}L'S$$

Since $S^{-1}L'S = S^{-1}(L - zz^T - \epsilon I)S$, then

$$S^{-1}(L - zz^T)S - \epsilon I = \begin{pmatrix} \lambda_1(G - e) & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2(G - e) & 0 & \cdots & 0 \\ 0 & 0 & \lambda_3(G - e) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_N(G - e) \end{pmatrix} - \epsilon I$$

The spectrum of L' becomes:

$$\tau_1 = -\epsilon, \tau_2 = \lambda_2 + \frac{\epsilon}{\frac{\alpha^2}{\epsilon+2} - 1}, \tau_3 = \lambda_3, \cdots, \tau_N = \lambda_N \quad (2.22)$$

Since our objective value is the second smallest in the sequence, the lower bound for $\lambda_2(G - e)$ is as follows:

$$\lambda_2(G - e) \geq \min \{ \tau_2 + \epsilon, \tau_3 + \epsilon \} \quad (2.23)$$

Substituting the values for τ_2 and τ_3 , we get:

$$\lambda_2(G - e) \geq \min \left\{ \lambda_2 + \frac{\epsilon}{\frac{\alpha^2}{\epsilon+2} - 1} + \epsilon, \lambda_3 + \epsilon \right\} \quad (2.24)$$

The best lower bound is therefore achieved by the choice of ϵ that makes both terms equal.

$$\begin{aligned}
\lambda_2 + \frac{\epsilon}{\frac{\alpha^2}{\epsilon+2} - 1} + \epsilon &= \lambda_3 + \epsilon \\
\lambda_2 + \frac{\epsilon\alpha^2}{\alpha^2 - \epsilon - 2} &= \lambda_3 + \epsilon \\
\lambda_3 - \lambda_2 &= \frac{\epsilon\alpha^2}{\alpha^2 - \epsilon - 2} - \epsilon \\
\lambda_3 - \lambda_2 &= \frac{\epsilon^2 + 2\epsilon}{\alpha^2 - \epsilon - 2}
\end{aligned} \tag{2.25}$$

For $\xi = \lambda_3(G) - \lambda_2(G)$,

$$\begin{aligned}
\xi (\alpha^2 - \epsilon - 2) &= \epsilon^2 + 2\epsilon \\
\alpha^2\xi - \epsilon\xi - 2\xi &= \epsilon^2 + 2\epsilon \\
\alpha^2\xi - 2\xi &= \epsilon(\epsilon + 2 + \xi) \\
\epsilon &= - \left[\frac{\xi + 2}{2} + \left(\frac{(\xi + 2)^2}{4} + \xi(\alpha^2 - 2) \right)^{\frac{1}{2}} \right]
\end{aligned} \tag{2.26}$$

Hence, a decrease in α decreases ϵ and increases the lower bound. Finally, combining the upper and lower bounds, we obtain the following bounds for algebraic connectivity after removing an edge:

$$\min \left\{ \lambda_2(G) + \frac{\epsilon\alpha^2}{\alpha^2 + (-2 - \epsilon)}, \lambda_3(G) + \epsilon \right\} \leq \lambda_2(G - e) \leq \lambda_2(G) - \alpha^2$$

As shown, a smaller α leads to a higher upper bound and also tends to increase the lower bound. This means that a smaller α leads to the minimal decrease in algebraic connectivity. In other words, we should remove an edge with the smallest $|u_i^{(2)}(G) - u_j^{(2)}(G)|$, that is an edge that connects two strongly connected vertices in G . Combining the removal and addition of edges, we obtain the following approach to rewiring edges such that algebraic connectivity increases the most:

1. Remove an edge such that $|u_i^{(2)}(G) - u_j^{(2)}(G)|$ is the lowest

2. Insert an edge such that $|u_i^{(2)}(G) - u_j^{(2)}(G)|$ is the highest

Or

1. Insert an edge such that $|u_i^{(2)}(G) - u_j^{(2)}(G)|$ is the highest

2. Remove an edge such that $|u_i^{(2)}(G) - u_j^{(2)}(G)|$ is the lowest

2.4 Network Models

This section reviews the three network models studied in this chapter:

1. Watts-Strogatz model
2. Gilbert Stochastic model
3. Barabási-Albert Scale Free model

2.4.1 Watts-Strogatz Model (WS)

The Watts-Strogatz model is constructed by interpolating between a regular ring lattice and a random network. Construction begins with a ring of N vertices, and each vertex is connected to its k nearest neighbors. Then, in a clockwise manner, vertex i is selected. The edge that connects to i 's nearest neighbor is randomly rewired with a probability of p (or left untouched with a probability of $1 - p$), considering the constraint that no self-loops or duplicate loops can exist. This procedure is repeated cyclically for each successive vertex until vertex i is once again selected. At this point, the edge that connects to i 's second nearest neighbor undergoes similar rewiring procedures. This cycle of vertex selection and rewiring recurs until the edge that connects all vertices i to their furthest neighbor is considered [50].

In the Watts-Strogatz model, the parameter p determines the level of randomness in the graph while maintaining the initial number of vertices and edges [34]. For intermediate values of p ,

Watts-Strogatz model produces a Small-world network which captures the high-clustering properties of regular graphs and the small characteristic path length of random graph models. Figure 2.1 shows three snapshots of graphs obtained for different values of p .

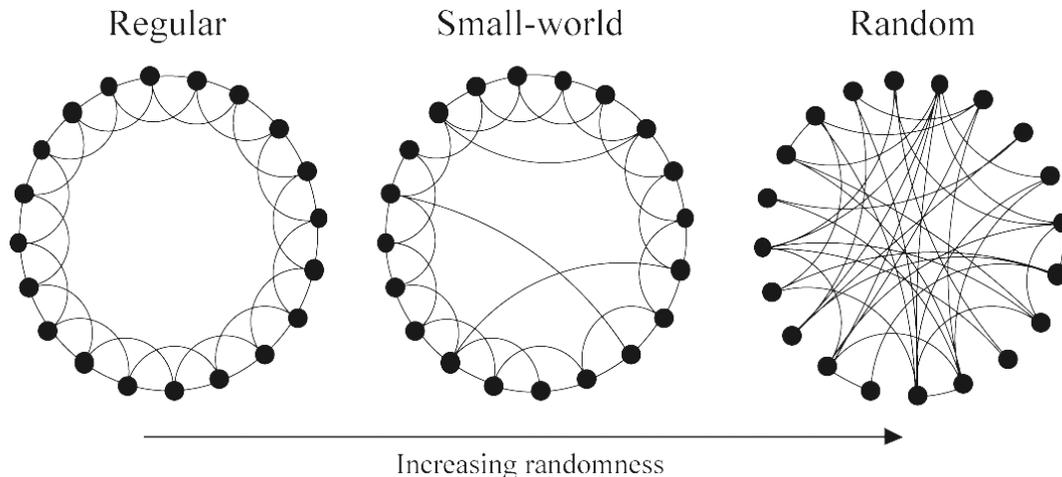


Figure 2.1: Construction of Watts-Strogatz model. For the regular graph $p = 0$. The random graph is obtained at $p = 1$ and for intermediate values of p , a Small-world network is realized [50].

For the Watts-Strogatz networks used in this chapter, we generated three networks with the respective sizes of $N = 100, 400, 800$ and a rewiring probability of 0.6 [51].

2.4.2 Gilbert Stochastic Model (Gi)

A random graph is obtained by random addition of edges between N vertices. Erdős-Rényi (ER) stochastic model is one of the most studied of these models. In the construction of an ER graph $G(V, E)$, $|E|$ edges are connected at random to $N = |V|$ vertices [34]. For this model, each of the $\frac{N(N-1)}{2}$ edges have an equal probability of being selected. However, this chapter considers the Gilbert stochastic model $G(V, p)$, a modified version of the ER model, in which edges are connected to vertices with a probability of p . As opposed to the ER model, the number of edges in a graph produced by the Gi model is not known in advance. Below are key properties of random graphs:

- The average node degree \bar{k} , such that $k = deg(v)$, determines the connectivity of the graph.

Therefore, if $\bar{k} < 1$, a disconnected component exists. At $\bar{k} = 1$, a phase transition occurs, and a giant component exists when $\bar{k} > 1$ [34].

- The node degree k exhibits a binomial distribution and thus, given N vertices and a probability of p ,

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}. \quad (2.27)$$

However, the model in this chapter was based on the poisson distribution, an approximation of the binomial distribution when the limit of N is large and $pN = \bar{k}$ [34].

$$P(k) = e^{-\bar{k}} \frac{\bar{k}^k}{k!} \quad (2.28)$$

- As k becomes large, the degree distribution decays exponentially

For this chapter we generated three networks of size $N = 100, 400$, and 800 with $p = 0.6, 0.05$, and 0.02 respectively, [51]. Figure 2.2 shows the node degree distribution for $N = 400$.

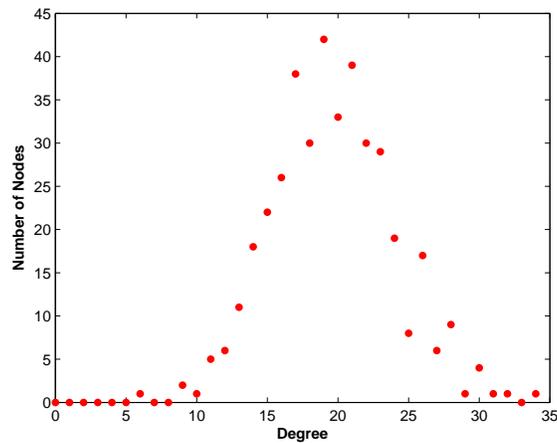


Figure 2.2: Node degree distribution for $N = 400$ and $p = 0.05$

2.4.3 Barabási-Albert Scale Free Model (BA)

Barabási-Albert Scale Free Models (also referred to as preferential attachment (PA) models) highlight a class of topologies associated with a heavy-tailed node degree distribution [52]. This distribution is also known as a power-law distribution. In particular, given a graph G with N vertices, the degree distribution is power-law if $P(k) \sim k^{-\sigma}$, where $\sigma > 1$ [36]. Furthermore, the power law distribution cuts-off at the maximum degree, $k_{cut-off} = n^{\frac{1}{\sigma-1}}$. The node degree distribution is defined as,

$$P(k) = \frac{n(k)}{N} \quad k = 0, 1, \dots, k_{max} \quad (2.29)$$

These networks pervade numerous real world domains. For example, within the sphere of social networks, an individual with few friends is more likely to form a new friendship with a more popular person. Likewise, new Internet websites will more likely establish ties with the most popular websites.

From their origin, BA models have been considered vulnerable to targeted attacks while robust to random failures [52]. This model constitutes popular vertices called “hubs,” which have a large number of neighbors compared to other vertices with few neighbors. The rules for construction are governed by two key principles of growth and preferential attachment. The initial number of vertices at construction must be greater than two and each vertex must have at least one neighbor. At each time step, a new vertex is added to the graph. The probability of attracting this new vertex is determined by the node degree of preexisting vertices. Thus, the higher the node degree of preexisting vertices, the higher their probability of attracting new vertices. The attachment probability is given by:

$$P(k_i) = \frac{k_i}{\sum_{j=0}^N k_j} \quad (2.30)$$

where $P(k_i)$ is the probability that a new vertex will connect to an existing vertex i with degree k_i [51].

For this chapter we generated three networks of size $N = 100, 400,$ and 800 [51]. Figure 2.3 shows the node degree distribution for $N = 400$.

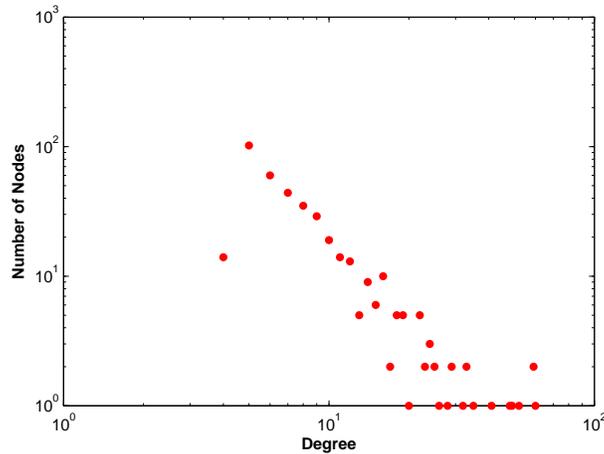


Figure 2.3: Node degree distribution for $N = 400$

2.5 Numerical Analysis for Edge Removal

In this section we generate three graphs which are representative of the three models presented in Section 2.4. We then use Algorithm 1 to realize the decrease in $\lambda_2(G)$ for all instances when an edge is removed.

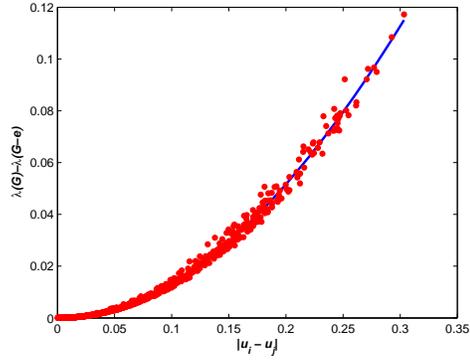
Figure 2.4 shows the decrease in algebraic connectivity for all realizations of an edge removal. These numerical results complement the analytical conclusions that removing an edge with the smallest absolute difference in the elements of the eigenvector (that is $|u_i^{(2)} - u_j^{(2)}|$ for vertices $i, j \in V$) corresponding to the second smallest eigenvalue (λ_2), tends to have the smallest decrease in algebraic connectivity. Furthermore, for these examples the coefficient of determination (R^2) shows that 99.4%, 99.5%, and 93.7% of the variation of $\lambda_2(G) - \lambda_2(G - e)$ for the Gi, WS, and BA networks, respectively, are accounted for by the polynomial relationship with $|u_i - u_j|$.

Algorithm 1 Algorithm for edge removal

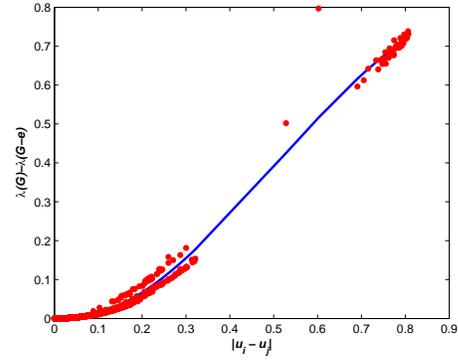
$A :=$ Adjacency matrix of graph G ; $N := |A|$
 $L :=$ The laplacian matrix of G
 $L' :=$ The laplacian matrix of $(G - e)$
 $R :=$ Matrix to store Lower bound, $\lambda_2(G - e)$, and Upper bound, such that e is edge (i, j)
for $i = 1$ to N **do**
 for $j = 1$ to N **do**
 if $(i \neq j$ and $A(i, j) = 1$ and $\lambda_2(G - e) > 0)$
 Remove e
 Compute L'
 Store $\lambda_2(G - e)$ in R
 Insert e
 Compute L
 Compute ϵ
 Store Lower and Upper bounds for $\lambda_2(G - e)$ in R
 end if
 end for
end for
Output R

2.6 Comparative Analysis of the Increase in Algebraic Connectivity via Edge Rewiring

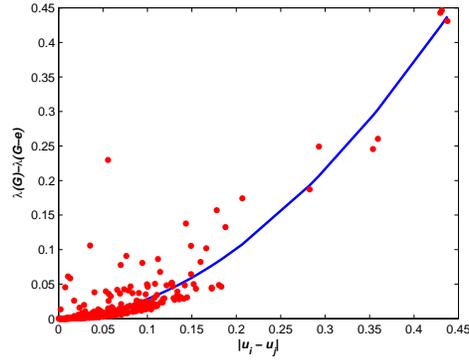
In this section, we compare the increase in algebraic connectivity through rewiring, for the three graph models presented in Section 2.4. In particular, for each network model, we first generate 10,000 networks, each with 100 nodes and 300 edges. For each network from the same model, we compute the initial value of algebraic connectivity (λ_2^i). We then rewire 7% of the edges and compute the final value of algebraic connectivity after rewiring (λ_2^f) and the difference between the final and initial values ($\lambda_2^f - \lambda_2^i$). This procedure is conducted for all 10,000 networks of a particular model and we averaged the results. Finally, we repeat this procedure for each network model. Figure 2.5 illustrates that for the Gi graphs, the average of λ_2^i is much lower than that of the BA and WS graphs, and the average of λ_2^f is also higher for Gi than for the other two graphs. With respect to the level of connectivity, this implies that networks from the BA and WS models tend to be more robust than that of the Gi model. Furthermore, if we compare the results of Figure 2.5 and Figure 2.6, we can deduce that graphs from the Gi model tend to have the highest gain in



(a) Watts-Strogatz model



(b) Gilbert stochastic model



(c) Barabási-Albert Scale Free model

Figure 2.4: Figures 2.4(a), 2.4(b), and 2.4(c) show the decrease in algebraic connectivity, as edges with the smallest α are removed for the Watts-Strogatz, Gilbert stochastic, and Barabási-Albert Scale Free networks, respectively. The coefficient of determination R^2 for the respective networks are 0.9954, 0.9935, and 0.9365, given a polynomial trend line with order 3.

algebraic connectivity for the proposed rewiring procedure.

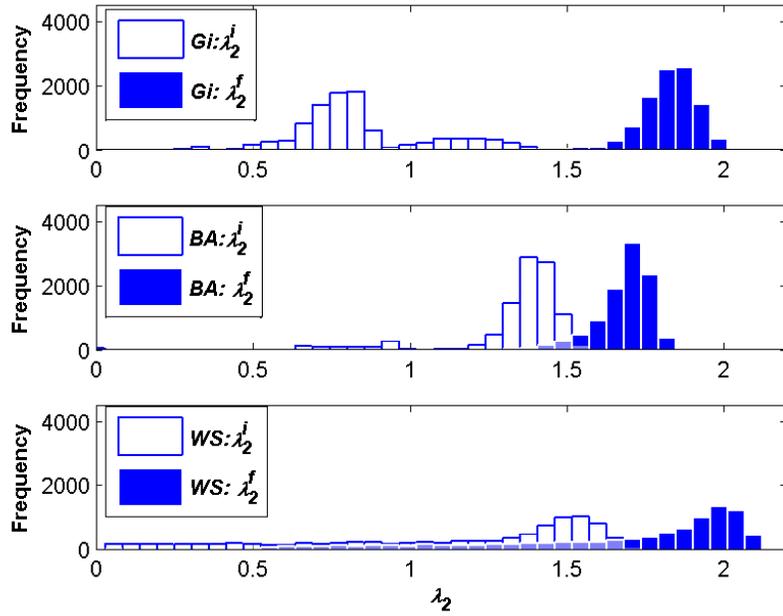


Figure 2.5: Distributions for the average values of λ_2^i and λ_2^f for Watts-Strogatz (WS), the Barabási–Albert Scale Free (BA), and the Gilbert stochastic (Gi) networks. $N = 100$ and $|E| = 300$. A transparency feature was utilized to visualize the overlap between distributions.

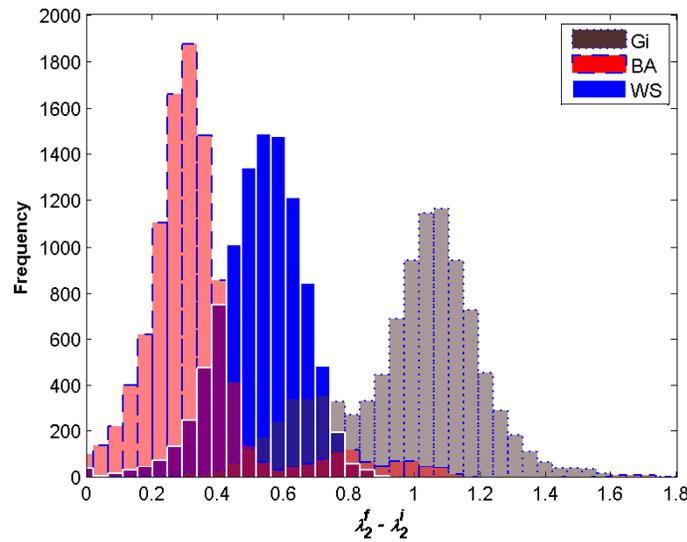


Figure 2.6: Distributions for the average values of $\lambda_2^f - \lambda_2^i$ for Watts-Strogatz (WS), the Barabási–Albert Scale Free (BA), and the Gilbert stochastic (Gi) networks. $N = 100$ and $|E| = 300$. A transparency feature was utilized to visualize the overlap between distributions.

2.7 Edge Rewiring to Maximally Increase Algebraic Connectivity

With the knowledge of which edge to remove to decrease algebraic connectivity the least and also which edge to insert to increase algebraic connectivity the most, we combine these two strategies to obtain Algorithm 2. In particular, Algorithm 2 rewires an edge by:

1. Removing an edge with the smallest α
2. Inserting an edge with the largest α

Similarly from Algorithm 2, if we reverse the removal/insertion order in the “while” statement such that first, $A(e_{\max}) = 1$ and second $A(e_{\min}) = 0$, we would rewire an edge by:

1. Inserting an edge with the largest α
2. Removing an edge with the smallest α

In the following simulations, Table 2.1 highlights the number of nodes and edges in the original nine graphs that were generated.

Networks	$N = 100$	$N = 400$	$N = 800$
Watts-Strogatz	1000	2000	4000
Random	2940	3925	6392
Barabási-Albert 1	451	1923	3913

Table 2.1: Elements of the table correspond to the number of edges for the specified network with size N

From Theorem 1, since $\text{trace}(L) = \sum_{i=1}^N \lambda_i(G) = 2|E|$, given a graph G with N vertices and $|E|$ edges, the magnitude of the eigenvalues increase with the $|E|$. This explains the huge variance in the magnitude of the eigenvalues in Figure 2.7 and 2.8 for the different classes of networks. As a result, in Figure 2.7 we expect the Gi network’s eigenvalues to be the highest (since it has the most edges), followed by that of the WS network, and the BA network. Similarly, in Figure 2.8

Algorithm 2 Algorithm for edge rewiring to maximally increase $\lambda_2(G)$

A :=Adjacency matrix of graph G
 L :=Laplacian matrix of G
 ψ :=% of edges to rewire
 e_{\max} := Edge $(i, j) \in E$ corresponding to α_{\max}
 e_{\min} := Edge $(i, j) \in E$ corresponding to α_{\min}
 $flag$:= Variable to ensure validity of while statement
for $i = 1$ to ψ **do**
 $flag = 0$
 Compute L
 Extract $u^{(2)}$, the eigenvector corresponding to $\lambda_2(G)$
 Compute α_{\max} and α_{\min}
 while $flag = 0$ **do**
 if $(e_{\min} \in G$ and $e_{\max} \notin G$ and $\lambda_2(G \setminus e_{\min}) > 0$)
 $A(e_{\min}) = 0$
 $A(e_{\max}) = 1$
 $flag = 1$
 else
 Find alternates for e_{\min} , e_{\max} , and $\lambda_2(G \setminus e_{\min})$
 end if
 end while
end for

we expect the G_i network to have the highest eigenvalues and the eigenvalues for the WS and BA to be comparable.

Figure 2.7 illustrates the propensity for algebraic connectivity to increase as 30% of the edges are rewired. The “*” denotes variation in the rewiring procedure where first, an edge with the smallest α was removed and second, an edge with the largest α was inserted (as opposed to the default rewiring procedure where first, an edge with the highest α is inserted and second, an edge with the smallest α is removed). As shown, both variations result in identical increases in algebraic connectivity. Finally, as shown in Figure 2.7 and more apparently in Figure 2.8, a rewiring threshold exists such that the algebraic connectivity is constant when this threshold is exceeded. Figure 2.8 in particular shows that, for the ER graph, there is no increase in algebraic connectivity beyond 8% rewiring. For the WS and BA networks, this phenomenon occurs at 20% rewiring.

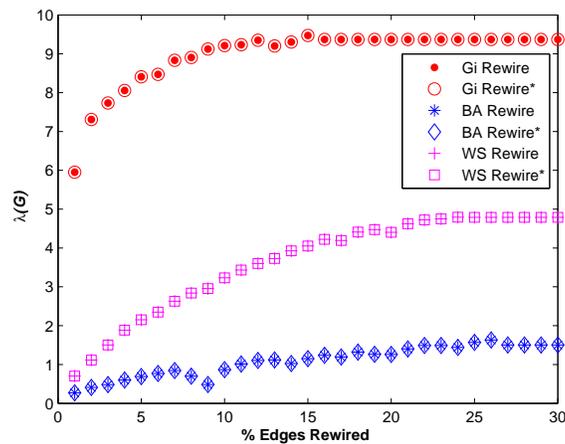


Figure 2.7: Increase in algebraic connectivity for Watts-Strogatz (WS), the Barabási-Albert Scale Free (BA), and the Gilbert stochastic (G_i) networks as edges are rewired by first inserting an edge then removing another. The “*” variation captures the results when rewiring is conducted by first removing an edge and then rewiring another. In this figure $N = 100$ and the values of λ_2 for 0% rewiring are 9.117, 2.757, and 42.834 for the WS, BA, and G_i networks, respectively.

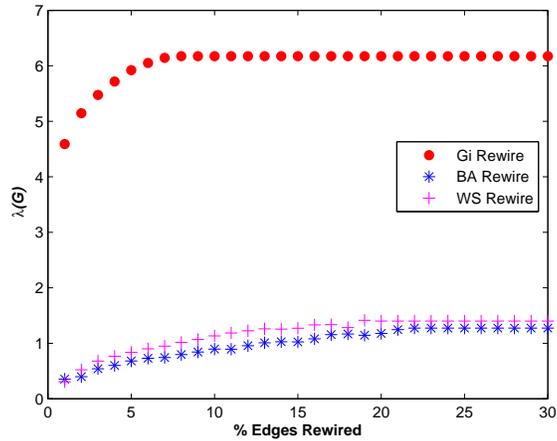


Figure 2.8: Increase in algebraic connectivity for Watts-Strogatz (WS), the Barabási-Albert Scale Free (BA), and the Gilbert stochastic (Gi) networks such that $N = 400$.

2.8 Rewiring vs Adding edges to Maximally Increase Algebraic Connectivity

In this section, we compare the results of rewiring to that of adding edges to maximally increase algebraic connectivity. For the addition of edges, we introduce Algorithm 3.

Figure 2.9 compares the increase in algebraic connectivity for rewiring and adding edges. It is immediately apparent that a large difference is present between rewiring and adding edges when the percentage of edges augmented (rewired/added) exceeds 5%. However, in a real-world scenario, the percentage of edges augmented can reasonably revolve around 1%, depending on the size and financial constraints of an organization.

2.9 Discussion

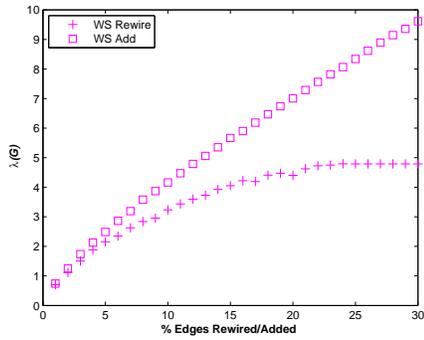
These results are important not only in the domain of graph theory but also in numerous complex networking domains such as the smart grid communication network, and even the transportation network. In the communication network domain, network engineers are constantly faced with the challenge of upgrading or, under certain circumstances, partially redesigning the network topology

Algorithm 3 Algorithm for edge addition to maximally increase $\lambda_2(G)$

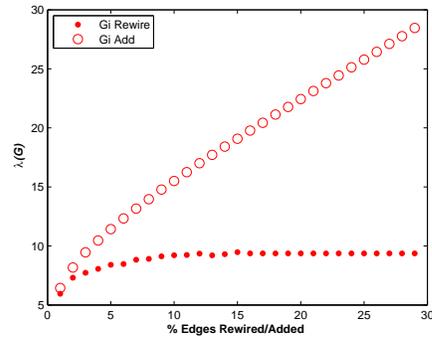
```
A :=Adjacency matrix of graph G
L :=Laplacian matrix of G
 $\psi$  :=% of edges to rewire
 $e_{\max}$  := Edge  $(i, j) \in E$  corresponding to  $\alpha_{\max}$ 
flag := Variable to ensure validity of while statement
for  $i = 1$  to  $\psi$  do
  flag = 0
  Compute L
  Extract  $u^{(2)}$ , the eigenvector corresponding to  $\lambda_2(G)$ 
  Compute  $\alpha_{\max}$ 
  while flag = 0 do
    if  $(e_{\max} \notin G)$ 
       $A(e_{\max}) = 1$ 
      flag = 1
    else
      Find alternates for  $e_{\max}$ 
    end if
  end while
end for
```

to increase connectivity. To accomplish such upgrades in most real-world cases, the number of edges to rewire or add is relatively small compared to the total number of edges in the network.

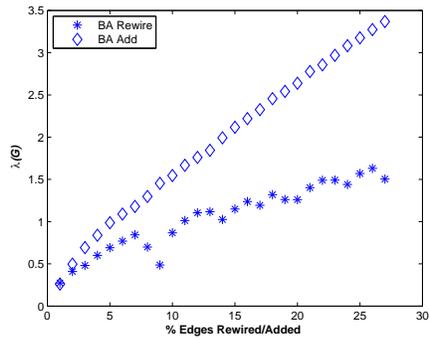
For $N = 100$, a 1% augmentation to the WS network is equivalent to augmenting 10 edges. For the Gi network, this equates to 29 edges, and for the BA network, this results in 5 edges. For the networks in Figures 2.9(a), 2.9(b), and 2.9(c), the increase in algebraic connectivity is comparable for both rewiring and adding edges if we are to consider a 1% augmentation. Similarly, Figure 2.10 compares the increase in algebraic connectivity for rewiring and adding 30 edges for $N = 800$. For such a small resolution in the number of edges augmented, results for adding edges are comparable to that of rewiring for all classes of networks. From a real-world perspective, this implies that for both rewiring and addition of edges, the number of edges required to disconnect a network is the same. Therefore, a solution that considers rewiring of edges is as robust as a solution that considers addition of edges. Thus, an organization can opt for either solution, depending on its economical and financial constraints.



(a) Watts-Strogatz model



(b) Gilbert stochastic model



(c) Barabási-Albert Scale Free model

Figure 2.9: Figures 2.9(a), 2.9(b), and 2.9(c) compare edge addition to edge rewiring to maximally increase algebraic connectivity in the WS, Gi, and BA networks, respectively.

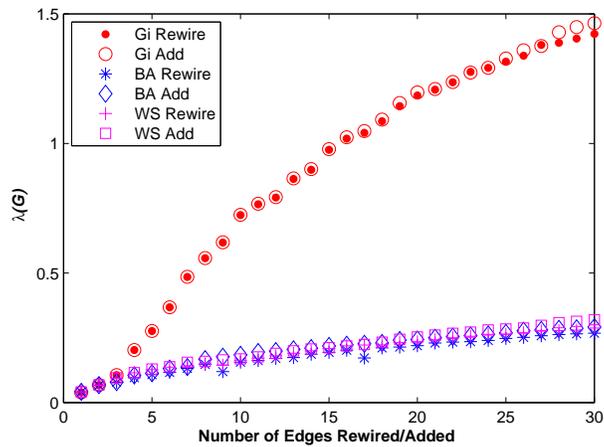


Figure 2.10: Comparing edge addition to edge rewiring to optimize algebraic connectivity in the WS, Gi, and BA networks for $N = 800$

2.10 Summary

To date, robustness in complex networks is an ongoing research effort. Among other topological measures, we use algebraic connectivity from spectral graph theory as our measure of robustness: the larger the algebraic connectivity, the more robust the network. In this chapter, we answer the question of, “Where should an edge be rewired to increase algebraic connectivity the most?” by dividing this question into two parts: “Where should an edge be removed to decrease algebraic connectivity the least?” and “Where should an edge be inserted to increase algebraic connectivity the most?” From our analytical results, we conclude that to decrease algebraic connectivity the least, we should remove an edge that connects two strongly connected vertices. Conversely, to increase algebraic connectivity the most, we should insert an edge between two weakly connected vertices. From our numerical results, we implement a rewiring strategy on three classes of networks that provides the maximal increase in algebraic connectivity and hence, the maximal increase in robustness of a graph.

From our simulations, we initially compare graphs from three classes of networks to determine the class that realizes the highest increase in algebraic connectivity. For an unbiased comparison, we set a constant number of nodes and edges for all networks and rewire a small percent of the edges. Our results reveal that graphs from Gilbert’s model (G_i) tend to have the lowest initial value for algebraic connectivity in addition to the highest gain in algebraic connectivity after rewiring. Subsequently, we compare the addition of edges to that of rewiring edges to maximally increase algebraic connectivity. We show that for edge augmentations (rewirings/additions) that exceed 5% of the network’s edges, the algebraic connectivity obtained when adding edges exceeds that obtained when rewiring edges. However, in real-world scenarios, such augmentations tend to be relatively small due to the non-negligible economical impact. In this case, the increase in algebraic connectivity is similar for both rewiring and addition of edges. From a real-world perspective, this implies that the number of edges required to disconnect the network is the same for both cases of rewiring or adding edges. Therefore, a solution that rewires edges is as robust as a solution where edges are added. Finally, our results illustrate that beyond a certain rewiring threshold ranging

from 8% to 20% for the graphs presented, algebraic connectivity is constant.

In this chapter, we used principles of graph theory to determine methods that yield robust communication network topologies. However, what is the impact on the characteristics of real-world networks when algebraic connectivity is maximized? Our next chapter, Chapter 3, addresses this question in detail.

Chapter 3

The Impact of Improving the Robustness of the Physical Communication Infrastructure for Transmission Operations in Smart Grids

In Chapter 2, we used principles of graph theory to determine methods that yield communication network topologies with high performance characteristics. However, what is the impact on the characteristics of real-world networks when algebraic connectivity is maximized? In response to this question, we conduct an analysis on the impact of algebraic connectivity maximization on the characteristics of the network topology. Subsequently, we conduct a simulative analysis on the impact of algebraic connectivity maximization on the reduction of traffic congestion. In this context, this chapter makes three primary contributions:

1. We illustrate that maximizing algebraic connectivity results in a more homogeneous network topology
2. We show that maximizing algebraic connectivity reduces the level of traffic congestion in a network

This chapter builds on the analytical and numerical results obtained from Chapter 2. The following outlines its organization. Sections 3.1 and 3.2 present a high-level description of the hybrid simulator that integrates the continuous-time behavior of a power model with the discrete-event

behavior of a communication network. In particular, Section 3.1 introduces the AC power system dynamic model and Section 3.2 reviews the basics of MPLS. In Section 3.3, we study the impact on the topology and traffic characteristics of networks when algebraic connectivity is increased. Finally, Sections 3.4 and 3.5 discuss the applicability and benefits of algebraic connectivity for real-world networks.

3.1 Power System Dynamics Model

Figure 3.1 gives a high level view of the grid today. The grid realizes the three fundamental roles of generation, transmission, and distribution. The generation region produces power to meet demands imposed by loads on the distribution side. Additionally, substations are dispersed throughout the transmission and distribution regions, facilitating control and monitoring functions such as servicing generators, distributing to customers, and boosting voltage.

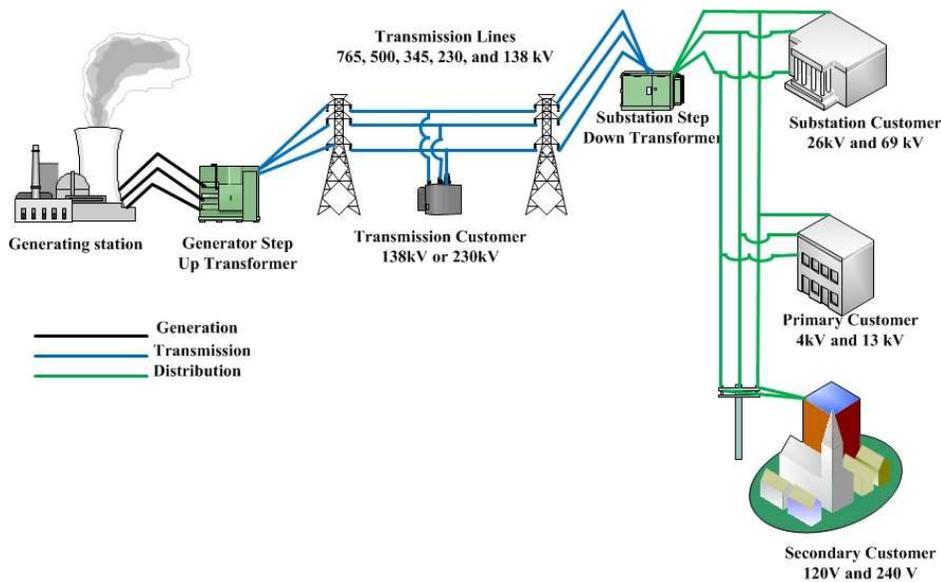


Figure 3.1: Panoramic view of today’s power grid [10]

Our model of the electric grid was constructed using THYME, an open-source C++ library which contains modules for simulating power grid dynamics and a framework for integration with discrete event models of communication networks [53]. The power system model implemented by

THYME was introduced in an earlier paper [54]; it is a simplified model for the study of electro-mechanical transients (see, e.g., [55, 56]). This model includes generators, their control systems for speed and voltage, and algebraic models of transmission circuits and loads.

The transmission circuits are modeled by assuming a voltage phasor at each bus. The voltage amplitude at the k th bus is V_k and the voltage phase angle is ϕ_k . At each bus, the load is modeled with a constant admittance. The generator, if present, is represented by a Norton equivalent circuit. The current injected into the network by a generator changes with time in accordance with its electro-mechanical dynamics. The impedance of the generator circuit is its complex synchronous reactance X_k .

The electro-mechanical dynamics of the generator at bus k are modeled with a set of differential equations that describe acceleration of the rotor due to power imbalance, speed control, and voltage control. These equations are

$$\dot{\omega}_k = \frac{P_{m,k} - P_{e,k}}{M_k} - D_k \omega_k \quad (3.1)$$

$$\dot{\theta}_k = \omega_k \quad (3.2)$$

$$\dot{c}_k = T_{1,k}(P_{s,k} - \omega_k/R_{g,k} - c_k) \quad (3.3)$$

$$\dot{P}_{m,k} = T_{2,k}(c_k - P_{m,k}) \quad (3.4)$$

$$\dot{E}_k = T_{e,k}(V_{s,k} - V_k) \quad (3.5)$$

The state variables in this model are the per unit deviation ω_k away from the network's synchronous speed (i.e., ω_k is deviation from the power system's synchronous speed; it is not the actual speed of the rotor), excitation voltage phasor $E_k \angle \theta_k$, mechanical power output $P_{m,k}$, demand for power $P_{e,k}$, and state c_k of the speed control system. The model's parameters are the voltage set point $V_{s,k}$, power set point $P_{s,k}$, droop setting $R_{g,k}$, inertia M_k of the rotor, resistance D_k to off-nominal speeds, and controller time constants $T_{e,k}$, $T_{1,k}$, and $T_{2,k}$. The first two equations are the swing equations, the second two equations model the speed controller, and the last equation models the excitation controller.

The output from the generator is the current it injects into the transmission system; its Norton

equivalent current I_k is

$$I_k = \frac{E_k \angle \theta_k}{X_k} \quad (3.6)$$

The generator has two inputs. First is the voltage phasor $V_k \angle \phi_k$ at its terminals, and second is the real demand for power

$$P_{e,k} = \text{Re} \left\{ \left(V_k \angle \phi_k \right) \left(\frac{E_k \angle \theta_k - V_k \angle \phi_k}{X_k} \right)^* \right\} \quad (3.7)$$

where Re is the real part of that complex quantity and $*$ denotes the complex conjugate.

Frequency regulation in this model is augmented by a hypothetical control system comprising sensors at the generators, a control center, actuators at the loads, and a communication network that links these three elements. Each generator is equipped with a frequency sensor that detects the generator's deviation $f_{c,k} = 60\omega_k$ from the nominal frequency of 60 Hz. The sensor detects changes at intervals of $f_{thres} = 0.0125$ Hz, and so reports a new value at $f_{c,k} = \dots, -f_{thres}, 0, f_{thres}, \dots$. At these instants, a packet carrying $f_{c,k}$ (hereafter called protection packets) is transmitted to the control center through the communication network.

The control center maintains an instantaneous average f_{avg} of the $f_{c,k}$ from the generators. The control center uses this information to change the admittance at each load by a fraction $K f_{avg}$ of its initial value. For this model, the choice of $K = 2$ yielded an acceptable control for the IEEE 118 bus model (see [57]). The effected load adjustment is

$$f_\alpha = K f_{avg} = \frac{K}{N_g} \sum_{k=1}^N f_{c_k} \quad (3.8)$$

where N_g is the number of generators in the system.

A new value for f_{avg} is computed each time the control center receives a protection packet. If the current value of f_α differs from the previously computed value, the control center transmits a protection packet to each load, enforcing an adjustment of their demands by f_α percent. In our model, f_α is arbitrarily restricted to 10% (e.g., to model the percentage of loads participating in the control scheme).

3.2 Communication Network Model For the Transmission Network

Figure 3.2 depicts the topology of the communication network model where Open Shortest Path First (OSPF) is the routing protocol within the inter-substation network and Label Switch Paths exist in the Multi-Protocol Label Switching (MPLS) core (MPLS is currently being adopted by utility companies). Table 3.1 provides the bandwidth measures and parameters for the corresponding media which can include Fiber and SONET (Ethernet over SONET, Digital Signal 1 'DS1', DS3, Optical Carriers such as OC48). We consider it a rare occurrence to add or rewire links in the the MPLS core. Hence, the maximization of algebraic connectivity considers links in the inter-substation network and the links between the inter-substation network and the MPLS core. The topology of the inter-substation network is similar to that of the underlying power network, except for the removal of buses co-located at the same substation. This reduces the number of communication nodes from 118 to 113 (with a total of 181 links).

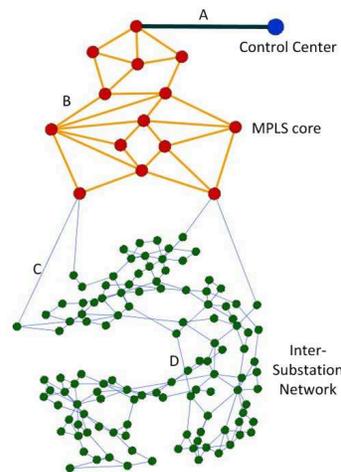


Figure 3.2: Model of the transmission communication network that is based on the IEEE 118 bus test case of the Power System's Test Case Archive [58]

Network Parameters	Bandwidth
A	1 Gbps
B	1 Gbps
C	500 Mbps
D	100 Mbps
Bitrate	400 Kbps
Packet Size	64 bytes

Table 3.1: *Network parameters for the Case Study*

For the simulation study, we consider two categories of traffic: protection (or control) and background. Section 3.1 provides a detailed description of our protection traffic scheme. To determine the bitrate for the background traffic, Table 3.2 shows four possible sub-categories of traffic and their corresponding bitrate. Since the aggregated bitrate is 400Kbps, a bitrate of 400Kbps was selected for each transmitting source.

Type of Traffic	Bitrate (Kbps)
Grid monitoring and control	7
Phasor measurement unit (PMU)	128
Intelligent fault management	10
Substation security video surveillance	255

Table 3.2: *Background traffic parameters for the transmission network [59, 60, 61].*

The background traffic profile is hypothetical and based on the ON/OFF model used to characterize Ethernet traffic [62, 63]. Since NS-3 does not model queuing delays for the MPLS module, the RTT presented is a combination of the queuing and transmission delays in the inter-substation network and the transmission delay within the backbone network. The transmission delay is evaluated as $\frac{\sigma_{ps}}{\sigma_{bw}}$ where σ_{bw} is the bandwidth input variable and σ_{ps} is the packet size.

3.3 Simulation Study

Detailed power results such as frequency profiles have been omitted, as we are primarily concerned with the impact of maximizing algebraic connectivity on the communication network (and not the power network). We first examine the impact to topological characteristics of a network

and, subsequently, the impact to characteristics of the traffic, as algebraic connectivity is increased. For the latter study, we explore Demand Response (DR) as an application where utility companies provide a cheaper billing rate to consenting customers. In return, these consumers allow utility companies remote access to control home appliances such as air conditioning units. In particular, during peak loading periods, these devices are powered off to reduce the load on generating resources, which in turn reduces the possibility of blackouts.

Considering results from Section 2.2 of Chapter 2, we investigate two scenarios for maximally increasing algebraic connectivity: 1) adding/rewiring links to/of the inter-substation network and 2) adding/rewiring links to/of the inter-substation network and the links between the inter-substation network and the MPLS core [21]. For both scenarios, the resulting number of links between the inter-substation network and the MPLS core is identical to that of the original network where links have not been added/rewired. For this reason, any increase in network performance is a result of the location where links are added/rewired and not a result of the number of added links.

3.3.1 Impact to Topological Characteristics of a Network as Algebraic Connectivity is Maximally Increased

We begin with the inter-substation network that has a topology identical to that of the underlying power grid and add/rewire links such that algebraic connectivity is maximally increased. Below is a list of all topologies under consideration:

- *LowBW*: The original 113 node topology
- $X\%Add_i$: The *LowBW* topology where $X\%$ of the total number of links are added to increase algebraic connectivity. i represents scenario 1 or 2.
- $X\%Rewire_i$: The *LowBW* topology where $X\%$ of the total number of links are rewired to increase algebraic connectivity.
- $0.3662Rewire_i$: The *LowBW* topology rewired such that algebraic connectivity is 0.3662

(i.e. the highest value obtained for rewiring the LowBW network for scenario 1. The corresponding networking in scenario 2 was rewired to achieve the same value).

To analyze these topologies, we present the following five topological metrics:

1. Diameter (D): Diameter is the longest shortest path between any source-destination node in a graph G .
2. Radius (R): Radius is the shortest of the set of all longest shortest paths from (or to) all nodes.
3. Characteristic path length (CPL): The expected shortest distance between two nodes.
4. Clustering co-efficient (C_{coe}): The clustering coefficient assesses how likely it is for a node and its neighbors to form a mesh.
5. Heterogeneity (H): For this metric, networks with an increasingly hub-like structure have a higher value [38].

Table 3.3 presents each topology and the resulting values for each of the corresponding metrics for scenario 1. From the original LowBW topology, as the percentage of links added/rewired increase, the values for all topological metrics tend to decrease. This was also the case for scenario 2.

Network	D	R	CPL	C_{coe}	H
<i>LowBW</i>	10	7	5.134	0.173	0.514
<i>1%Add₁</i>	9	6	4.833	0.167	0.498
<i>5%Add₁</i>	8	6	4.485	0.140	0.465
<i>10%Add₁</i>	8	5	4.200	0.110	0.420
<i>10%Rewire₁</i>	8	6	4.493	0.062	0.399
<i>0.3662Rewire₁</i>	8	6	4.358	0.034	0.365

Table 3.3: *Impact on topological characteristics of a network as algebraic connectivity is maximally increased*

Analytical results from Section 2.2 in Chapter 2, along with the results of Table 3.3, indicate that increasing algebraic connectivity tends to remove hubs and results in a topology that exhibits

a more homogeneous node degree (where node degree is the number of links connected to a node). In the following section, we insert these topologies in a hybrid simulator and study impact to the characteristics of traffic.

3.3.2 Impact to Characteristics of Traffic as Algebraic Connectivity is Maximally Increased

At simulation time 1, we fail generator 6 on bus 49 and observe the network response as both protection and background traffic are transmitted through a given topology. Each node is configured with multiple interfaces that facilitate interconnection with other nodes. Table 3.4 shows the total number of bytes lost, averaged over the simulation time for all interfaces of every node for the *LowBW*, *1%Add₁*, *1%Add₂*, *5%Add₁*, *1%Add₂*, and *10%Add₁* networks, respectively. For omitted networks, no bytes were lost. The networks are ranked from highest to lowest byte lost and, as observed, there is a decrease in the number of bytes lost as algebraic connectivity is increased. Most notable is that the *LowBW* network, whose topology is identical to that of the underlying power infrastructure, exhibits the highest loss of bytes, implying that a topology which may be ideal for the power network may not be ideal for the communication network. However, though corresponding results for scenario 2 are omitted, the number of bytes lost for scenario 2 is much less than scenario 1, demonstrating that adding/rewiring links considering a larger fraction of the network results in a higher performing network as compared to adding/rewiring links only to the inter-substation network.

Network	Total Average Bytes Lost
<i>LowBW</i>	136300.4
<i>1%Add₁</i>	82798.1
<i>1%Add₂</i>	10461.4
<i>5%Add₁</i>	28971.4
<i>5%Add₂</i>	7973.13
<i>10%Add₁</i>	4284.5

Table 3.4: Average number of bytes lost at each interfaced, totaled over all interfaces for scenarios 1 and 2

The following figures bolster results shown in Table 3.4. In particular, Figures 3.3, 3.4, 3.5,

and 3.6 illustrate the average number of bytes lost at each interface of every node for the *LowBW*, $1\%Add_1$, $5\%Add_1$, and $10\%Add_1$ inter-substation networks, respectively. For the $10\%Rewire_1$ and $0.3662Rewire_1$ networks, no bytes were lost. These figures demonstrate that as algebraic connectivity is increased, there is a decrease in the number and height of peaks corresponding to the number of bytes lost. Most notably is that the *LowBW* network in Figure 3.3, whose topology is identical to that of the underlying power infrastructure, exhibits the highest lost of bytes.

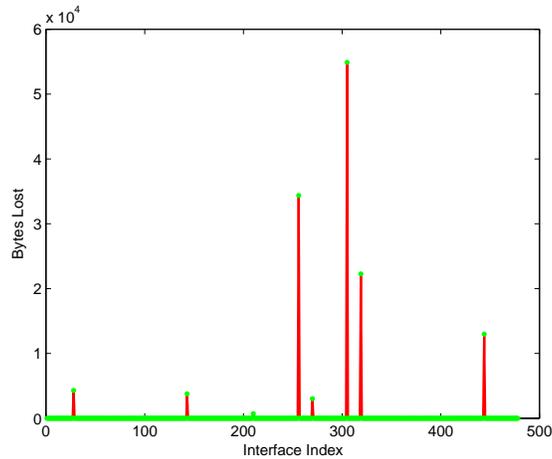


Figure 3.3: Number of bytes lost at each interface for all nodes in the *LowBW* network

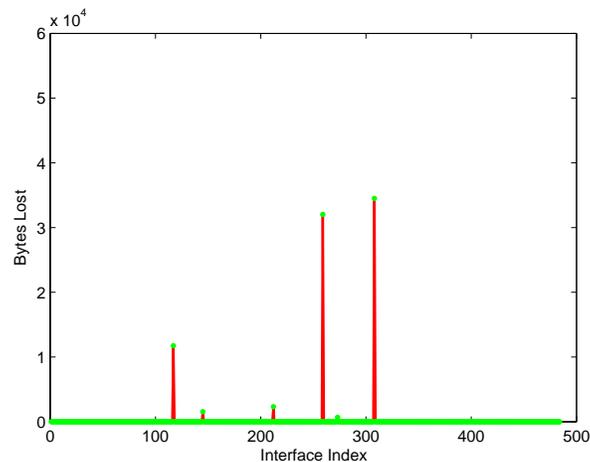


Figure 3.4: Number of bytes lost at each interface for all nodes in the $1\%Add_1$ network

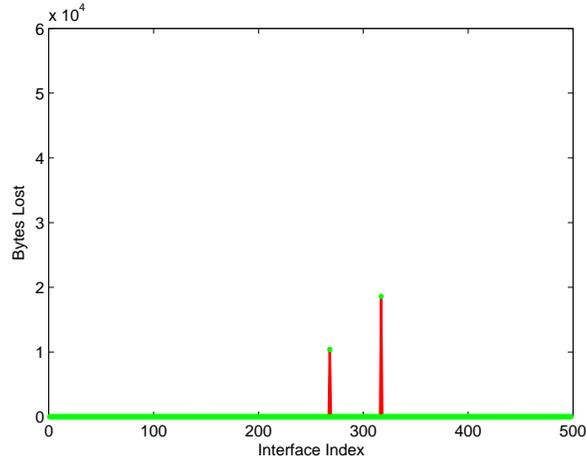


Figure 3.5: Number of bytes lost at each interface for all nodes in the 5%Add₁ network

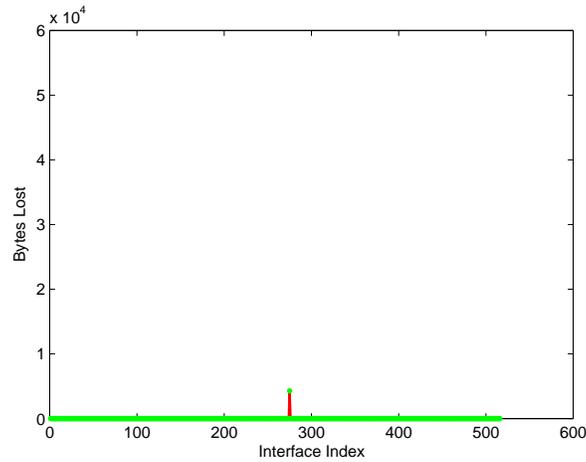


Figure 3.6: Number of bytes lost at each interface for all nodes in the 10%Add₁ network

For the second scenario, Figures 3.7 and 3.8 show the number of bytes lost at each interface. We omit the plots for 10%Add₂, 10%Rewire₂, and 0.3662Rewire₂ as the number of bytes lost was negligible. From these figures, we recognize a similar trend as in scenario 1: increasing algebraic connectivity decreases the number and height of peaks corresponding to the number of bytes lost. However, the number and height of “byte lost” peaks for scenario 2 is much less than that of scenario 1. Intuitively, this demonstrates that adding/rewiring links considering the multi-layer network results in a higher performing network as compared to adding/rewiring links only

considering the inter-substation network.

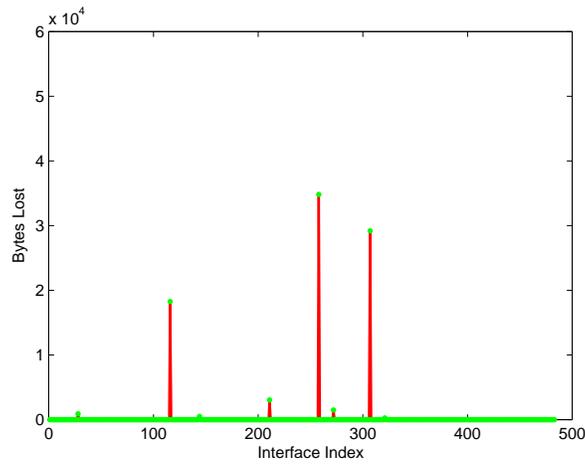


Figure 3.7: Number of bytes lost at each interface for all nodes in the 1%Add₂ network

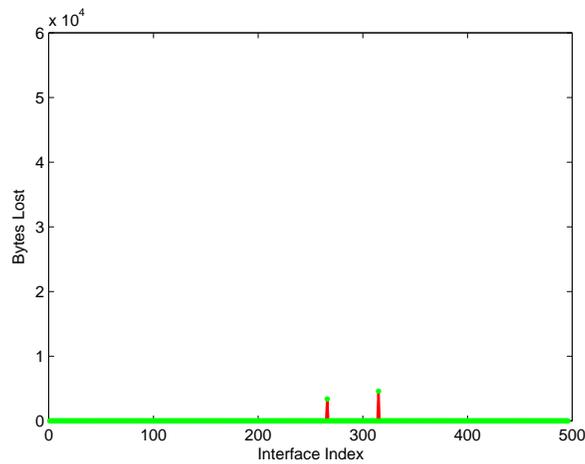


Figure 3.8: Number of bytes lost at each interface for all nodes in the 5%Add₂ network

Figure 3.9 captures the number of source-destination flows through every interface for the respective inter-substation network (a modified version of the betweenness metric in graph theory). A high number of flows through any interface indicates an ill-designed network that tends to be congested. The *LowBW* network can be considered a benchmark for the worst designed network as there exists four interfaces with peak flow values that exceed all other networks. However, as 1% of the links are added (i.e. 1%Add₁), the number of peaks has dropped from four to two, with

a decrease of about 150 flows. Furthermore, as we increase the number of links by 5% and 10%, these peaks no longer exist. One interesting observation is that rewiring 10% of the links results in less flows per interface when compared to adding 10% of the links. One possible explanation is that when rewiring, we tend to remove links between strongly-connected nodes and add links between weakly connected nodes. This removal tends to reroute the flow of traffic on links that are possibly congested. As opposed to rewiring, there is a low probability that adding links between poorly connected nodes will reroute the flow on links that are already congested. Though the resulting number of flows per interface is lower when we consider scenario 2, we observe similar reductions in the number of flows per interface as links are added/rewired.

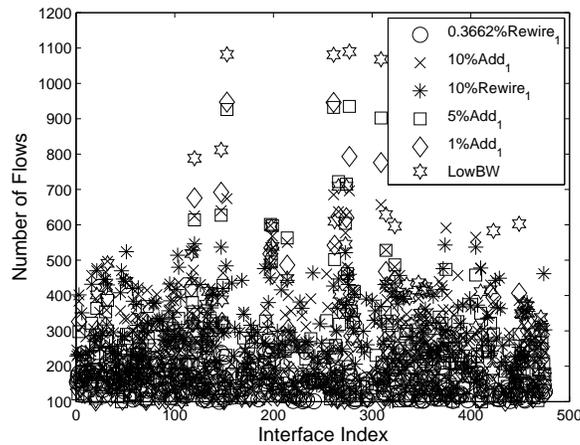


Figure 3.9: Number of source-destination flows traversing the interfaces of each node in the respective inter-substation network (i.e. the betweenness of an interface)

NS-3 provides statistics such as mean delay and the number of packets transmitted for each source-destination flow. For all flows which arrive at the CC, we sum the number of bytes received (σ_{rx}) and used Equation 3.9 to obtain the throughput as follows:

$$Tp = \frac{8\sigma_{rx}}{2 * 10^6} \tag{3.9}$$

where the factor of 8 converts the number of bytes to bits, 2 in the denominator is the interval at which throughput measurements are recorded, and 10^6 in the denominator converts bits to

Megabits. This results in a throughput measurement in Mbps. Furthermore, we use Equation 3.10 to calculate the RTT as follows:

$$RTT = \frac{\sum_{i=0}^{\zeta} \beta_i}{\zeta/2} \quad (3.10)$$

where β is the mean 2-way delay extracted from each flow, and ζ is the total number of flows between every source-CC node pair. In the denominator, ζ is divided by 2 because each flow is unidirectional. Table 3.5 shows each network and the corresponding values for the network metrics. In particular, the throughput (Thpt), RTT, and PLR values registered at the CC, are averaged over the simulation time. Max Util is the maximum utilization for the simulation time of 20s and the *HighBW* network is identical to the *LowBW* network except that all bandwidth capacities are 1Gbps. This network serves as the optimal benchmark for all other networks.

Ranked from highest to lowest throughput, the results from Table 3.5 show that adding/rewiring links to increase algebraic connectivity tends to improve network characteristics for scenario 1. In particular, an increase in throughput and a decrease in RTT and PLR occurs. Similar trends were observed for the results of scenario 2, in addition to the fact that the performance measures for scenario 2 exceeded that of scenario 1.

Network	Thpt (Mbps)	RTT (ms)	PLR	Max Util
<i>HighBW</i>	63.25	0.096	0	0.782
0.3662 <i>Rewire</i> ₁	52.48	1.110	0.184	1.002
10% <i>Add</i> ₁	51.62	1.052	0.184	1.002
10% <i>Rewire</i> ₁	47.37	1.195	0.271	1.002
5% <i>Add</i> ₁	47.07	1.416	0.275	1.002
1% <i>Add</i> ₁	33.92	1.779	0.465	1.002
<i>LowBW</i>	29.78	2.117	0.526	1.002

Table 3.5: Impact on network characteristics as algebraic connectivity is increased in scenario 1

In Figure 3.10, we compare the throughput and PLR for the two scenarios. From Figure 3.10, all PLR vs Throughput values fall between the measurements for the LowBW and HighBW benchmark networks. Most importantly, networks from scenario 2 tend to out perform their counterparts in scenario 1. For example, 1%*Add*₂ realized a lower PLR and higher throughput than

its counterpart $1\%Add_1$. Once again, this indicates that considering a larger fraction of the network when adding/rewiring links tends to result in a higher performing network, as opposed to adding/rewiring where we consider a smaller fraction of the network (i.e. inter-substation network). These general trends are also reflected when we consider the RTT vs the throughput in Figure 3.11.

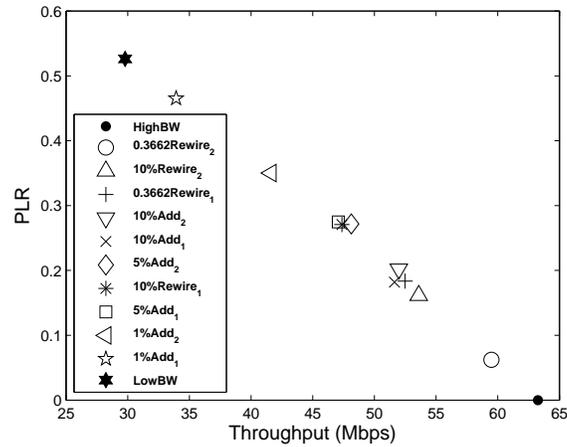


Figure 3.10: Comparison of throughput and PLR for networks in scenario 1 and networks in scenario 2

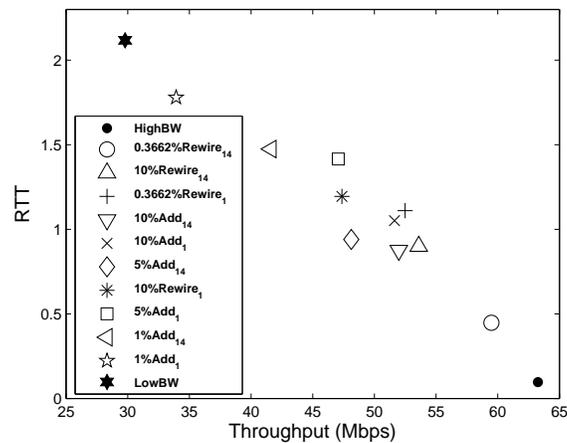


Figure 3.11: Comparison of throughput and RTT for networks in scenario 1 and networks in scenario 2

3.4 Discussion

Algebraic connectivity is a spectral measure that defines the well-connectedness of networks. Therefore, the greater the fraction of links that must be removed to fragment a network, the greater the value of algebraic connectivity. The question is asked, “How does algebraic connectivity translate into real-world networks?” As we increase the number of links added/rewired, a network becomes more homogeneous. Over time, this homogeneity provides multiple shortest paths and, as a result, reduces congestion in the network.

As utilities are in the initial phase of deploying communication infrastructure, algebraic connectivity can be used as a tool to design cost-effective networks. The results of this chapter first demonstrate that a topology which may be ideal for the power network, may not be ideal for the communication infrastructure. Second, we illustrate that rewiring links can produce the same performance as adding links to a network. Links can be rewired or added to achieve a particular PLR and throughput. However, there is a threshold such that further rewiring does not improve the network’s performance.

3.5 Summary

In this chapter, we use spectral analysis to obtain strategies to add and rewire links such that algebraic connectivity is maximally increased. We used these strategies to transform a communication network, identical to the power grid network, into multiple instantiations such that the resulting networks seek to improve on the characteristics of the original. Each topology was grouped into one of two scenarios. For the topologies in each scenario, we first analyzed the topological impact as algebraic connectivity is increased. We then inserted each topology in a hybrid simulator to study impact to the network characteristics of traffic as algebraic connectivity is increased.

The topological results demonstrate that adding/rewiring links creates a more homogeneous network. Network traffic results illustrate that a network which may be ideal for the power network may not be ideal for the communication network. A comparison of the PLR, RTT, throughput, and

betweenness measurements for both classes reveal that adding/rewiring links to a larger fraction of the network results in a higher performing network. In some cases, rewiring links provide similar performances to adding links. For utility companies at the design phase of deploying a communication infrastructure, this implies that it may be more cost-effective to rewire a network, than to continue adding links.

In this chapter, we demonstrated the correlation between the performance of the communication network and the robustness of the physical communication infrastructure. In our next chapter, Chapter 4, we evaluate and compare the performance of network technologies which will be deployed on the physical communication infrastructure.

Chapter 4

Simulative Comparison of Multiprotocol Label Switching and OpenFlow Network Technologies for Transmission Operations

Currently, utility companies are gravitating towards MPLS as their backbone communication technology for two main reasons: it supports Virtual Private Networks (VPNs) and Traffic Engineering. To provide these services, multiple protocols, such as Open Shortest Path First (OSPF) and Resource Reservation Protocol (RSVP), are implemented by the network. However, with MPLS, the addition of new network services requires the implementation of new protocols on network routers. Consequently, routers and other network equipment may require extensive reconfiguration and exhaustive testing that may cause intermittent service interruptions.

On the contrary, OpenFlow, which has gained support from network providers such as Microsoft, Google, Amazon and equipment vendors such as NEC, Juniper, and Cisco, is a highly modular networking technology that provides the functionality of MPLS and the ability to isolate network traffic generated by different services and applications [13, 14, 15]. In particular, changes to network services require a simple change in the OpenFlow controller deployed on the network operating system. Furthermore, with OpenFlow new services are not tied to extensions of existing protocols. This is unlike MPLS, for which new services must be implemented in each router and often times tied to an existing service; for example, RSVP-Traffic Engineering (TE) is tied to RSVP.

In addition to these advantages, OpenFlow’s ability to isolate network traffic ensures that failure of an experimental protocol, service, or application does not affect other experiments or hinder production traffic. In the same way, different classes of traffic in the smart grid can be isolated for Quality of Service (QoS) guarantees. For these reasons, OpenFlow may provide a more capable backbone communication technology that is overall less expensive than MPLS.

This chapter explores, via simulation, the potential for using OpenFlow network technology to support production and research traffic for a smart grid on the same communication network, and to reduce the cost of adding new services to an operational network. The Toolkit for Hybrid Systems Modeling & Evaluation (THYME) and Network Simulator 3 (ns-3) simulation tools (see [64, 53]) were used to compare the performance of MPLS and OpenFlow networks in the context of a hypothetical smart grid application. The goal of this application is to regulate frequency by monitoring generator speeds, transmitting these speeds to a control center where they are processed, and then issuing actuation commands to increase or decrease the power consumed by loads. In these experiments, it is shown that OpenFlow performs as well as MPLS with respect to regulating frequency and quantity of load required for regulation.

This chapter is organized as follows: Section 4.1 reviews the basics of MPLS and OpenFlow and introduces the two models for our communication backbone network, one for MPLS and another for OpenFlow using Intelligent Switch Controllers (ISCs). These models were integrated with the power model described in Section 3.1 of Chapter 3 to realize the prototype of our hybrid smart grid model. Section 4.2 presents simulations of each model. Finally, Section 4.3 discusses the benefits and shortcomings of OpenFlow and highlights future work.

4.1 Simulative Communication Network Model

NS-3 was selected as our simulator for the communication network. NS-3 is an open-source, discrete-event simulator primarily developed for academic and research initiatives. NS-3 is extended by creating new modules with the C++ language, which facilitates seamless integration with THYME. Furthermore, the NS-3 development community provides a rich set of real-world,

network component models that include the MPLS routers and OpenFlow switches employed in this work.

4.1.1 MPLS

As an overlay technology, MPLS provides IP services over legacy TDM devices and integrates multiple transport technologies such as fiber, SONET, and Digital Microwave. Furthermore, MPLS realizes the NERC Critical Infrastructure Protection (CIP) standards, in part, due to its non-routable nature [12, 11]. Specifically, MPLS routers on a Label Switch Path (LSP), excluding the end-point Label Edge Routers (LERs), forward packets based on the MPLS label, and not the IP address to port mappings found in the Routing Information Base (RIB) which is common to routers. MPLS provides fast, efficient forwarding of IP packets by adding a new label to the header of a frame.

4.1.2 OpenFlow

The fundamental components of the OpenFlow Architecture include a flow table, secure channel, and OpenFlow protocol such that the control and data paths are separate [65, 66, 67, 68, 69]. Below is a high level description of the OpenFlow mechanisms:

1. The OpenFlow (OF) controller uses the OF protocol to install flow space rules in the flow table of the OF switch preemptively or at run time.
2. As flows from substation communication systems arrive at the OF switch, they are checked against a list of flow space rules in the flow tables.
3. If a packet from a stream does not match any rule in the flow table, the first packet of this stream is encapsulated and transmitted to the OF controller for further evaluation.
4. After evaluation, the OF controller installs a new rule for this type of packet and all subsequent packets encounter similar actions without visiting the OF controller.

OpenFlow Model

The above functionality may be implemented with Learning OF switches or Intelligent OF switches. A Learning Switch Controller (LSC) creates a dynamic table mapping source IP address to switch port for each ingress packet of a flow. Subsequent packets are forwarded by the OF controller to their destination if the destination IP address in these packets are found in the table. An obvious disadvantage of the Learning Switch is that every packet on egress at the switch is forwarded to the OF controller, and for this reason the OF controller becomes a bottleneck to the network.

The Intelligent Switch Controller (ISC) includes the basic features of the LSC. Additionally, it employs a flow installation mechanism that inserts rules and corresponding actions in the flow table of the switch [70, 71, 72, 73, 74, 75]. Unlike the Learning Switch, the Intelligent Switch has mechanisms in the OF controller to mimic the functionality of the Label Switch Paths (LSPs) in MPLS.

Two important OpenFlow parameters that could potentially degrade network performance are the “idle timeout” (the time such that if no packets are received, a flow is removed from the flow table) and the “hard timeout” (the time such that all flows are removed from the flow table whether packets are in route or not). These parameters do not exist in MPLS. Since the communication nodes at the generators are transmitting protection data at millisecond intervals, the “hard timeout” parameter was disabled to avoid unnecessary removal of flows from the flow table.

Link and Traffic Model

The link and traffic model presented here is similar to that presented in Section 3.2 of Chapter 3 with added consideration for the OpenFlow component of the hybrid model. For the substation network, communication nodes are located at every substation such that interconnectivity of these nodes are identical to that of the underlying power network. Furthermore, two categories of traffic were considered: protection and background. Section 3.1 of Chapter 3 provides a detailed description of our protection traffic scheme. Background packets are generated by each node using a hypothetical ON/OFF model originally developed for Ethernet traffic [62, 63]. Data rates for the

background data are taken from experiments reported in [59, 60, 61]. Table 3.2 summarized these experimental data rates.

The ON/OFF model for generating background traffic works as follows. The first transmission of background traffic by a node occurs at a time selected from a uniform distribution with mean of 1.75s. The node transmits 64 byte packets at a rate of 400 Kbps (i.e., approx. 780 packets per second) for 2s. Transmission of data then stops for 1.5s, which forces the expiration of flows in the OpenFlow switch if its idle timeout is 1s or less. When the 1.5s pause is over, transmission starts again and this pattern is repeated.

Unlike the node module in NS-3, the MPLS and OpenFlow modules in NS-3 do not model queuing delays. For this reason, the delay presented subsequently is a combination of delay within the substation network and delay within the backbone network. The backbone network only considers the transmission delay: $\frac{\alpha_{ps}}{\alpha_{bw}}$ where α_{ps} is the packet size and α_{bw} is the bandwidth. However, the substation network considers both the transmission delay and the queuing delay at each node.

4.2 Simulation Studies

Two simulation studies were considered for the power system model and control scheme described in Section 3.1 of Chapter 3, using the following technologies in the backbone communication network:

1. MPLS routers
2. OpenFlow Intelligent Switches

For each simulation study, the IEEE 118 bus case shown in Figure 4.1 was first considered and then subsequently, the 300 bus case [58]. To evaluate the 300 bus communication network, the 300 bus case was substituted for the 118 bus case, maintaining four links between the inter-substation network and the two MPLS/OpenFlow core nodes. All subsequent plots have subscripts of 118 and 300 in the legend to distinguish between results for the 118 and 300 bus case, respectively. The subsequent sections refer to a network with ISCs as OpenFlow.

The initial disturbance in each simulation occurs at $t = 1$, when generator 6 on bus 49 of the IEEE 118 bus is disconnected from the power system. For the 300 bus system, generator 4 on bus 79 is disconnected. The controllers for speed and voltage at the generators are disabled, and so control of the loads is the only means for damping the frequency excursions that begin after the disconnections.

Using this scenario, two sets of parameters were considered for the network by including or omitting background traffic and by varying the bandwidth of the links labeled C and D in Figure 4.1. Table 4.1 shows the two parameterizations that are used. For each parameterization, the effectiveness of MPLS and OpenFlow were compared for implementing load control. Specifically, comparisons were made for final voltages in the power network, the average frequency variation for the generators, the amount of load that is adjusted, and throughput and latency in the communication network. In all figures where an "idle timeout" of 1 or 2 seconds was not specified, the result for either time out values were identical.

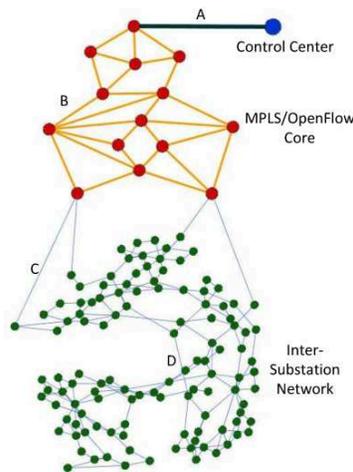


Figure 4.1: *Communication network for IEEE 118 bus case of the Power System’s Test Case Archive.*

4.2.1 Simulation study #1

This section considers a high performance network where the bandwidth on all links is set to 1Gbps and no background traffic exists. This simulation provides a benchmark for all subsequent

Network Parameters	BW:SS1	BW:SS2
A	1 Gbps	1 Gbps
B	1 Gbps	1 Gbps
C	1 Gbps	500 Mbps
D	1 Gbps	100 Mbps
Background traffic	no	yes

Table 4.1: *Network parameters for simulation study 1 (SS1), and 2*

experiments; the network is optimal, with a bandwidth of 1Gbps at all tiers, and protection data is the only traffic on the network.

Figure 4.2 compares the initial and final bus voltages for this scenario for the 118 bus system. For this figure, as well as all other figures in Section 4.2.1, the "idle timeout" for the OpenFlow simulations did not affect the results as protection packets were continually streamed through the communication network. As noted, the voltage profile using MPLS is comparable to that of OpenFlow (OF). For these two cases, only generator 6 fails. From this figure, "Final Voltage: No Load Control*" coincides with the final voltage profile for both MPLS and OpenFlow cases. However, without load control, generators 5, 6, and 7 go offline within the first second of the initial failure. Simulation Study #2 reflected similar results. For the 300 bus system, generator 4 failed with load control but generators 4 and 40 failed with no load control. Though these plots have been omitted, voltage profiles for both MPLS and OpenFlow were comparable for both simulation studies.

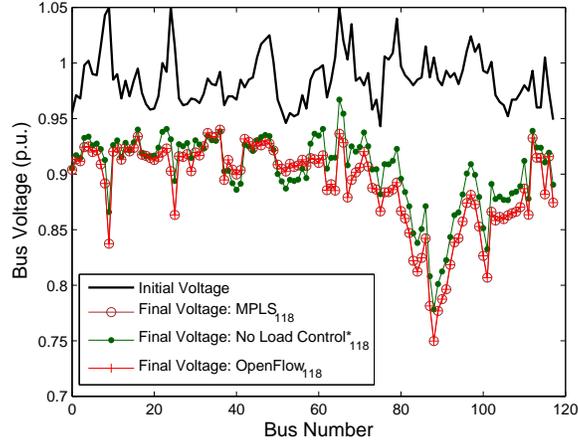


Figure 4.2: Comparison of initial and final voltage profiles for a high bandwidth OpenFlow/MPLS communication network where only protection traffic exists.

When generator 6 of the 118 bus case fails, the frequency begins to decrease and demand gradually exceeds supply. Figure 4.3 shows that without load control, a failure of generators 5, 6, and 7 brings the frequency deviation dangerously near the threshold of $\pm 1\%$ of the nominal frequency set for generators to go offline [76].

For the cases where load control is used, though generator 6 is offline, the remaining 18 generators are able to supply sufficient power to loads to stabilize the frequency. This result is expected as generator 6 contributes only 4.7% of the total power.

From Figure 4.3, the origin of the graph corresponds to our nominal frequency of 60Hz. As illustrated by the MPLS and OpenFlow results, reducing the demand imposed by the loads prevents a drop of frequency. In particular, as the frequency begins to drop, the load adjustment scheme is executed and a slight increase is noted at approximately 2.5s. Eventually, near 5s, the system becomes stable and only one generator (generator 6) goes offline.

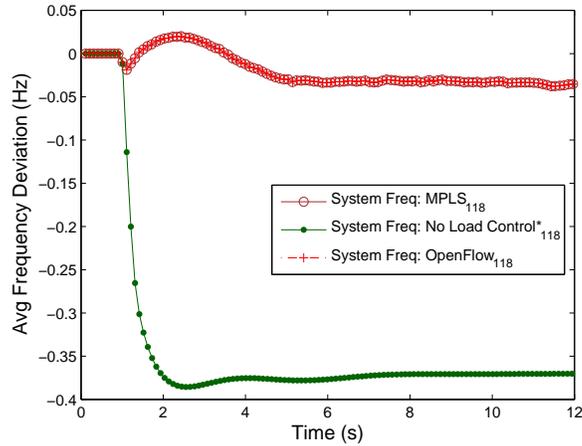


Figure 4.3: Comparison of resulting frequency profiles for the MPLS and OpenFlow networks

For the 300 bus system, the frequency deviation is smaller since generator 4 constitutes 0.5% of the total power provided by 64 generators. As shown in Figure 4.4, there is a dip in the frequency at 2s, but the system quickly stabilizes.

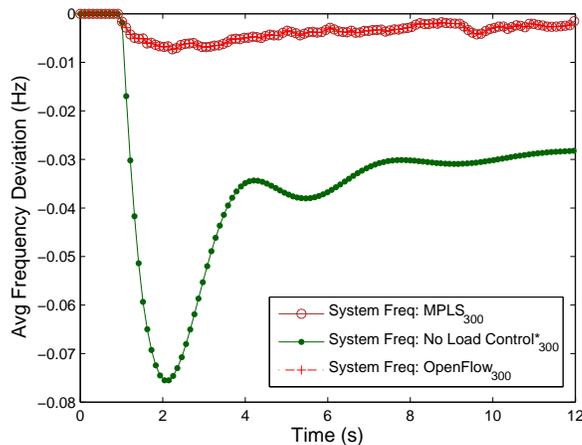


Figure 4.4: Comparison of resulting frequency profiles for the MPLS and OpenFlow networks

Figure 4.5 shows the value of f_α imposed by the control center on loads. As described in Section 3.1 of Chapter 3, Equation 3.8, f_α is restricted to 0.1. However, in the graphs depicting f_α , the values have been normalized such that 0.1 is equivalent to 100% load adjustment. As shown in Figure 4.5, there is an initial “ramping-up” throughout the first 5s for the 118 bus case. From 6s

onwards, though, the graph becomes stable for both MPLS and OpenFlow network configurations. For the 300 bus case, load shedding occurs throughout the first 2s. Subsequently, there is a gradual increase in power consumed by loads as the system stabilizes.

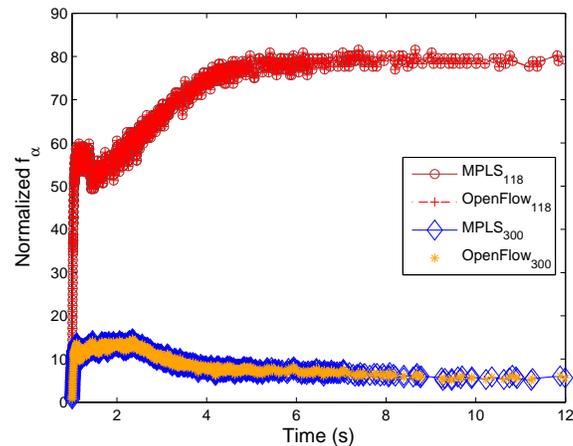


Figure 4.5: Comparison of resulting rate of load adjustment for MPLS and OpenFlow networks

NS-3 has the concept of a (unidirectional) flow which contains statistics for every transmission between source-destination node pairs. For each flow, these statistics include the associated mean delay, number of packets transmitted, and number of packets lost. For all flows which arrive at the control center, Equation 3.9 gives the throughput value.

Figure 4.6 portrays the throughput measurement for this simulation study. Both the x and y axis are plotted in log scale to demonstrate that the throughput does not go to zero. One interesting observation is that the initial throughput at the onset of the simulation is highest and decreases as the system stabilizes. This is caused by the rapid adjustment of the loads in the first part of the simulation; comparing Fig. 4.5 and Fig. 4.6 shows that the throughput and rate of adjustment are closely related. For this simulation study, the Packet Loss Ratio (PLR) was 0.

Equation 4.1 calculates the Round Trip Time (RTT) as follows:

$$\frac{\sum_{i=0}^{\zeta} \beta_i}{\zeta/2} \quad (4.1)$$

where β is the mean 2-way delay extracted from each flow, and ζ is the total number of flows between every source-control center node pair. In the denominator, ζ is divided by 2 because each

flow is unidirectional. For Simulation study #1, the RTT value only considers protection traffic (i.e. background traffic is excluded). In particular, RTT was 0.294ms for the MPLS backbone network and 0.232ms for the OpenFlow backbone network when the 118 bus case was considered. For the 300 bus case, the RTT for MPLS and OpenFlow backbones were 0.312ms and 0.358ms, respectively.

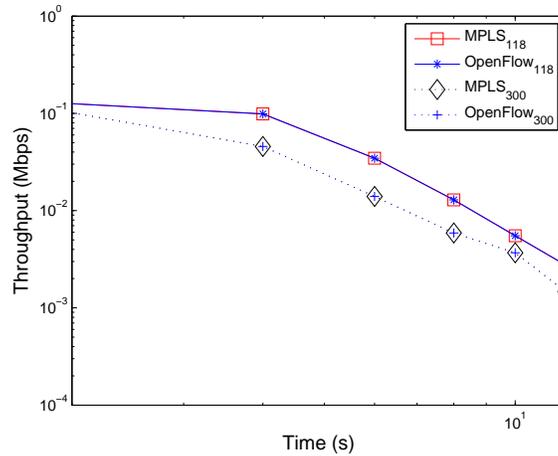


Figure 4.6: Comparison of throughput (traffic generated and received) for a high bandwidth MPLS/OpenFlow communication network where only protection traffic exists

4.2.2 Simulation study #2

Simulation study #2 repeats the MPLS and OpenFlow simulations from simulation study #1 but adds background traffic and reduces the bandwidth of links C and D. For the 118 bus case, Figures 4.7 and 4.8 show small and almost insignificant changes in frequency and load adjustment profiles for the range of "idle timeout" values considered; OpenFlow simulations with the "idle timeout" set to 1s and 2s yield identical results.

Figure 4.7 shows the average frequency in simulations with OpenFlow and MPLS. Most importantly, though the graph depicts rapid "zig-zag" behaviors, the load shedding facilitated by both MPLS and OpenFlow, depicted in Figure 4.8, was sufficient enough to contain the frequency deviation well within the 1% threshold set for generators to transition offline.

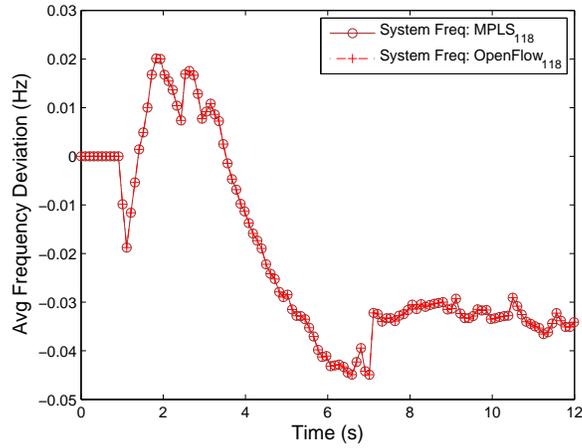


Figure 4.7: Comparison of resulting frequency profiles for MPLS and OpenFlow communication networks

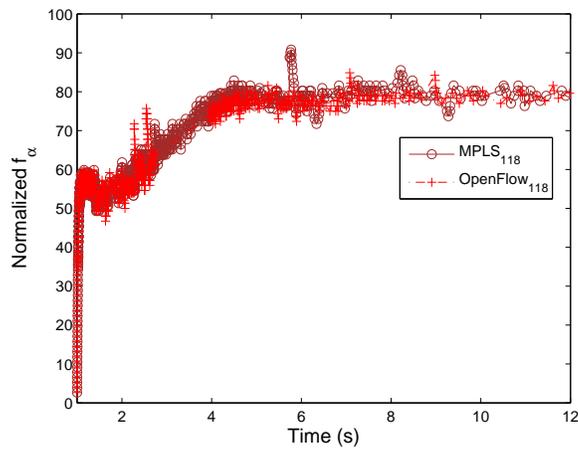


Figure 4.8: Comparison of resulting rate of load adjustment when using an MPLS/OpenFlow network

Figure 4.9 shows the frequency profile for the 300 bus case. As opposed to the smaller 118 bus case, deviations were observed between 2s to 6s, with the MPLS and OpenFlow-2s networks stabilizing after 6s. However, there is a noticeable increase in the deviation for the OpenFlow-1s network which can be attributed to congestion and constant removal and re-installation of flows every second. The OpenFlow-1s network approaches a stabilized state after 12s.

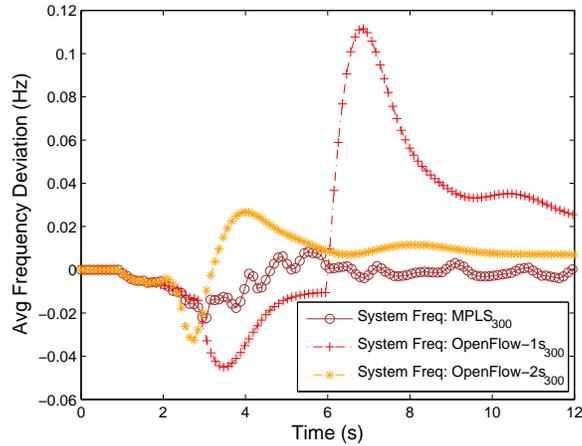


Figure 4.9: Comparison of resulting frequency profiles for MPLS and OpenFlow communication networks considering the 300 bus system

In Figure 4.10, rapid changes in load adjustment measurements transmitted from the control center to the loads are apparent. Unlike simulation study #1, the OpenFlow-1s network rapidly sheds load 3s after the OpenFlow-2s network. Most importantly, at 6s when the OpenFlow-1s network rapidly load sheds, both the MPLS and OpenFlow-2s networks have stabilized. Once again, these delays can be attributed to congestion and the 1s timeout parameter to remove idle flows from the OpenFlow switches.

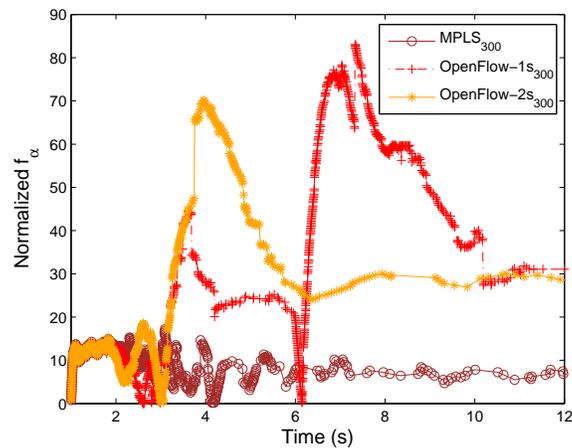


Figure 4.10: Comparison of resulting rate of load adjustment when using an MPLS/OpenFlow network

Figures 4.11 and 4.12 show the throughput for both the OpenFlow and the MPLS backbone networks. As the number of sources of background traffic increase, the throughput increases to a peak value. Then, as initial background sources reach the end of the 2s “ON” interval, the peak decreases. This summation of uniform distributions results in a series of normal distributions with peaks occurring at every 3.5s interval. In Figure 4.11, the throughput measurement for the MPLS Low BW network was identical to that for the OpenFlow network where the flows did not expire (i.e. OF Low BW-2s). Most importantly, given a randomized packet generation start-time from a uniform distribution with a mean of 1.75 and a time out value of 1s, the likelihood that all flows will expire simultaneously decreases. As a result, the throughput for the OF Low BW-1s network (where flows expire) occasionally decreased by a fraction of 1Mbps from the scenario where flows did not expire. For the larger 300 bus case in Figure 4.12, with a decrease in bandwidth, increase in congestion, and the given timeout parameters, the maximum throughput deviation between the MPLS Low BW, OpenFlow-1s, and OpenFlow-2s was approximately 5Mbps.

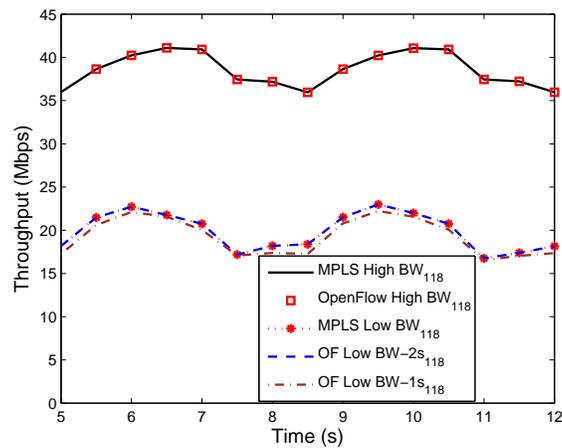


Figure 4.11: Comparing throughput for a low bandwidth MPLS/OpenFlow communication network where background and protection traffic exists. MPLS and OpenFlow High BW are the benchmark throughput values for the high BW network in simulation study #1 with the addition of the background traffic profile for simulation study #2. MPLS Low BW represents a Low BW MPLS network and OF Low BW-Xs represents the simulation of an OpenFlow network with a timeout value of X seconds.

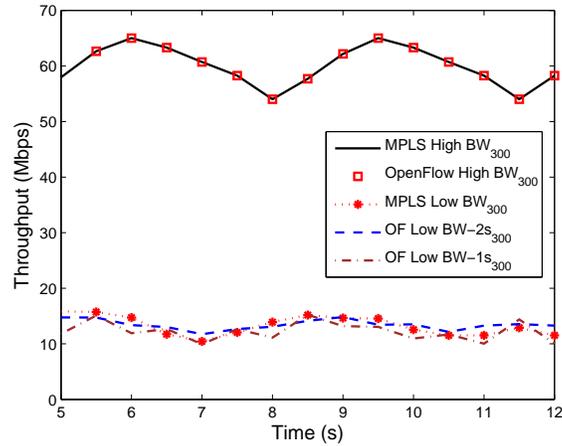


Figure 4.12: Comparing throughput for the 300 bus case.

The substation communication network used NS-3’s Open Shortest Path First (OSPF) routing where all nodes are configured with a routing table that contains an entry for every destination. Furthermore, these nodes were configured with the Drop-Tail-Queue model characterized by the variable MaxPackets, where MaxPackets was set to 1000 for the 118 bus case and 10,000 for the 300 bus case. The Drop-Tail-Queue realizes a First In, First Out (FIFO) queuing scheme that drops tail-end packets when the queue is full.

Considering the MPLS High BW network for the 118 bus case, the PLR and RTT were 0 and 0.18ms, respectively. For the MPLS Low BW and OF Low BW-2s networks, the average PLR and RTT were approximately 0.41 and 3.44ms, respectively. Finally, for the OF Low BW-1s network, the average PLR and RTT were 0.44 and 3.91, respectively. With the exception of the MPLS High BW network, the PLR values are substantially high as packets are dropped at the queues. For the 300 bus case, the PLR and RTT values were much higher but similarly comparable for the networks considered. For example, the PLR and RTT for the MPLS High BW network was 0.59 and 21.6ms, respectively. For the OpenFlow High BW network, the PLR and RTT was 0.61 and 22.7ms, respectively. In real-world networks, network engineers will not design networks to exhibit such high PLR values.

Finally, though the results are comparable, MPLS provides the best performance characteris-

tics. However, with MPLS the LSPs are pre-configured, allowing traffic arriving at the egress of all routers to be forwarded immediately. On the contrary, paths are constructed during transmission of packets for OpenFlow: a sub-optimal configuration for operating an OpenFlow switch. However, OpenFlow allows installation of flows/paths prior to transmission of packets, thereby operating more closely as an MPLS router and bridging the small performance gap between the two technologies.

4.3 Summary

This work is the first phase towards demonstrating that a relatively inexpensive OpenFlow switch can perform as well as an MPLS switch when used for control in the smart grid. In particular, a hybrid model that integrates the continuous time behavior of the power grid with the discrete event behavior of the network was developed. Our results indicate that setting the OpenFlow timeout parameter to expire before the completion of a transmission, can decrease the throughput and increase the PLR and RTT of a network. However, the resulting throughput, PLR, and RTT is comparable to that of an OpenFlow network where flows do not expire or MPLS low bandwidth network with similar traffic demands. An OpenFlow network that is configured such that the timeout parameter exceeds the completion of a transmission performs comparably to its MPLS counterpart under similar network constraints. Furthermore, preemptive installation of flows in OpenFlow can realize an even higher level of performance. Finally, since OpenFlow supports all features of MPLS, it can seamlessly co-exist with MPLS devices.

In this chapter, we demonstrated via simulation that an OpenFlow network performs as well as MPLS. However, the current OpenFlow hardware does not readily support MPLS. In any case, can we use commercially-available OpenFlow hardware to provide similar mechanisms as MPLS? Chapter 5 addresses this issue.

Chapter 5

Software Defined Networking (SDN) in GENI: Experimental Evaluation of OpenFlow Technology for Smart Grids

It has been demonstrated that OpenFlow can provide similar services as MPLS using Open VSwitch software switches [13]. However, to date, the current OpenFlow hardware does not readily support MPLS. In any case, can we use commercially-available hardware in GENI to provide MPLS-like functionalities? To answer this question, we contribute the following:

1. An OpenFlow controller that implements an automatic fail-over mechanism and traffic engineering services such as auto-route, load balancing, flow preemption, auto-bandwidth, and fast re-route.
2. A Demand Response (DR) smart grid application that transmits traffic created by cyber physical systems

The structure of this chapter is outlined as follows: Section 5.1 builds on the introduction by providing necessary background and state-of-the-art for networking solutions within utility companies. We review various research projects that consider simulations, emulations, and real-time communication network implementations and experiments for the smart grid. Section 5.2 presents a high level overview of the smart grid model. In particular, details are provided for the Electro-Mechanical prototype and overall smart grid prototype. We also present a brief review of

the OpenFlow architecture. Section 5.3 describes the three experiments conducted and provides the resulting throughput, frequency response, and load shed performance results. Section 5.4 presents a discussion on the applicability of this work in the real world and setbacks encountered. Finally, Section 5.5 discusses the benefits and shortcomings of current hardware and highlights the future direction of this work.

5.1 Background and Related Work

Utility companies have been reluctant in adapting to the changing demands in communication networks to support increasing smart grid tools and applications for several reasons. On one hand, they are tasked with providing reliable and secure communications to clients thus, being almost surely driven towards communication solutions that have been well vetted over the years. For utilities unable to maintain their own private networks, service providers are subcontracted to support communication network services. On the other hand, research in designs of alternate networking architecture will unlikely be deployed without a demonstration of an actual prototype under realistic conditions. Deployment of a prototype in the production setting of a utility company will almost surely be discarded, not only due to the tradition of utilizing tried and tested solutions, but also due to the steep fines of millions of dollars per day charged by NERC should a utility be in violation of any standards.

A driving force towards deploying innovative ideas can be attributed to the three phases of:

1. developing thorough models
2. exhaustively testing these models on simulators
3. exhaustively testing these models in real-time

To date, phase one has gained considerable attention as several models exist that consider the continuous dynamics of the Power Grid through ordinary differential equations [77, 78]. However in phase two, few models that integrate both Power and network models are prevalent in literature

[79]. One of the first attempts at this class of hybrid simulators can be attributed to EPOCHS. The EPOCHS framework federates close-source discrete-event and continuous time packages through a mediating control agent [80]. In particular, both power and communication system simulations run simultaneously but independently, until they arrive at a predetermined synchronization point. At this point, the simulations pause while a mediating agent accesses the internal data of both simulators and executes a data exchange routine between simulators. Subsequently, the simulators resume executing until the next synchronization point. As a first-cut effort, EPOCHS' contribution was the foundation of other such simulators. However, it has been proven that this approach introduces timing errors due to the difficulty in selecting synchronization points. This framework can produce behaviors which are independent of the actual model [81].

An improvement to the EPOCHS framework demonstrates the removal of synchronization point dependence by using the global scheduler of the communication network simulator [81]. Specifically, the power system dynamic simulation is divided into several discrete events distributed over the simulation time-line. Events from both simulators are entered into the global scheduler of Network Simulator 2 (NS2), which allows instant response to events.

A second approach is that of the open source ADEVS modules, that models continuing dynamics of power systems through the DEVS framework. In DEVS, continuous time dynamics are represented by discrete-events using state-detection mechanisms such as zero crossings [82]. Discrete events from both simulators are implemented by the global scheduler of NS2, as was done in the previous approach. Finally, the ADEVS approach not only closely approximates the costly GE Positive Sequence Load Flow (PSLF) package used in the previous approaches, but by virtue of its open source origins, the ADEVS modules are free for research purposes and provide seamless integration between continuous time and discrete event simulators.

A third approach is an improvement to the ADEVS approach. In this approach, the Toolkit for Hybrid Systems Evaluation and Modeling (THYME) was integrated with the Network Simulator 3 (NS-3) simulation tools (see [64, 53]) to compare the performance of MPLS and OpenFlow networks in the context of a hypothetical smart grid application [83].

Another class of simulations have been conducted using the mininet simulation framework to demonstrate that OpenFlow can provide similar services to MPLS using an OpenFlow control-plane and the same push, pop, and swap behavior used in the MPLS data-plane [84, 85]. Furthermore, researchers demonstrated a low-cost MPLS Label Switch Router (LSR) using NFPGAs that realizes an implementation of Label Distribution Protocol (LDP) using the Quagga routing suite [14].

In phase three, several experiments exist, including a research project which seeks to develop technologies to integrate fixed (hydro, flywheel, and compressed air) and mobile (batteries in cars) storage to the power grid [86]. However, these projects rely on existing network architectures where innovation is restricted to features enclosed in the “box” [87, 88, 89, 90, 91].

For this reason, the Global Environment for Network Innovation (GENI) at the U.S. National Science Foundation provides researchers network resources, scale, realism, and control necessary to deploy prototypes and evaluate new networking architectures. To date, the OpenFlow specification 1.1.0 supports MPLS; however, the current commercial hardware does not. In this project, we integrate the current network hardware in GENI and the power resources of Kansas State University as a smart grid prototype where automatic fail-over and traffic engineering services are provided.

5.2 Power and Smart Grid Model

Figure 5.1 illustrates a high level view of a smart grid where a network provides the communication and control to the generation, transmission, and generation components of the grid. This model provides visibility to the Control Center (CC) and allows customers to interact with the system.

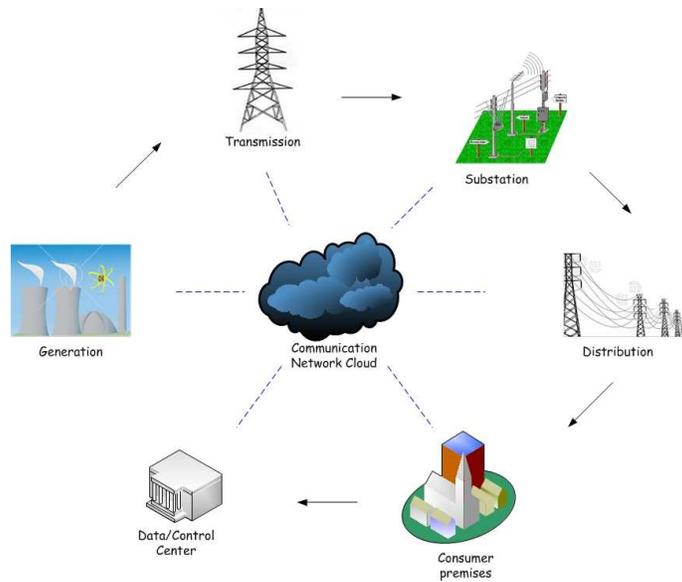


Figure 5.1: *Model of the smart grid*

5.2.1 The Electro-Mechanical Model

The power system test-bed in Figure 5.2 displays a 4-bus system consisting of three synchronous generators (G1, G2, and G3), three transmission lines, and three loads. During normal operation, G3 at bus 1 produced 95W. A 3-phase autotransformer was placed at bus 1 to reduce the voltage from 208V to 138V to accommodate the equipment's voltage requirements. A 3-phase diode bridge rectifier and capacitors were placed on the low side of the transformer to form a 160V DC bus. The DC bus had 2 loads: an 11W fixed load and an Agilent 6063B variable electronic load operating in constant resistance mode. Nominally, the electronic load's resistance was set at 200Ω (120W). A 90W load and a generator (G2) operating at 120W were connected to bus 2. G3 at bus 3 normally produced 65W and there was no load at bus 3. The buses were connected in a loop with inductive transmission lines. Each transmission line had a reactance of $j1.2241\Omega$. In order to test the load shedding action, the circuit breaker on generator 3 (CB G3) was opened, disconnecting G3 from the system. The loss of G3 was enough to depress the system frequency by at least 3Hz.

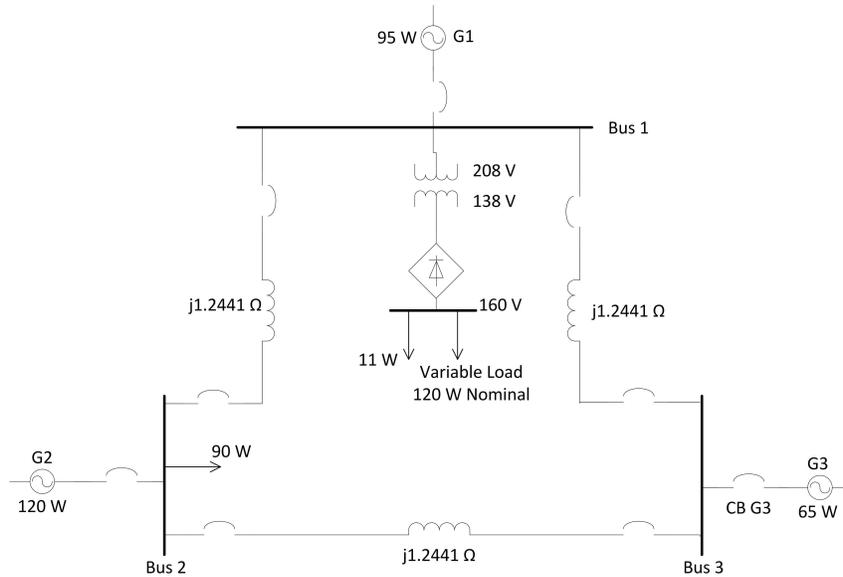


Figure 5.2: *Prototype of the power grid*

5.2.2 The Smart Grid Prototype

Figure 5.3 provides details of the smart grid prototype. Resources are divided into two groups: resources of K-State and resources of GENI. At K-State, an analog to digital converter converts the analog voltage to its digital equivalent. This digital signal is transmitted to a micro-controller that counts the width of each pulse to provide an estimation of its period. The period is then serially transmitted to ksuHost1. A generator agent (GA) at ksuHost1 transmits 1 period measurement, for every 15 samples received from the micro-controller (i.e. protection traffic), through the network to the Control Center agent (CCA) in GENI. Assuming the frequency has deviated from the nominal value of 60Hz, the CCA transmits load shed measurement to the load agent (LA) at ksuHost3. The LA communicates to the Agilent 6063B variable load through a GPIB connection to adjust the load accordingly such that a frequency of 60Hz is maintained. Additionally, a loop topology exists in the GENI core for redundancy and dual-homing purposes, and an OpenFlow controller residing at the control center, provides the control plane control for all OpenFlow switches in GENI. ksuHost2 generates streams of background traffic to the host at the Control Center.

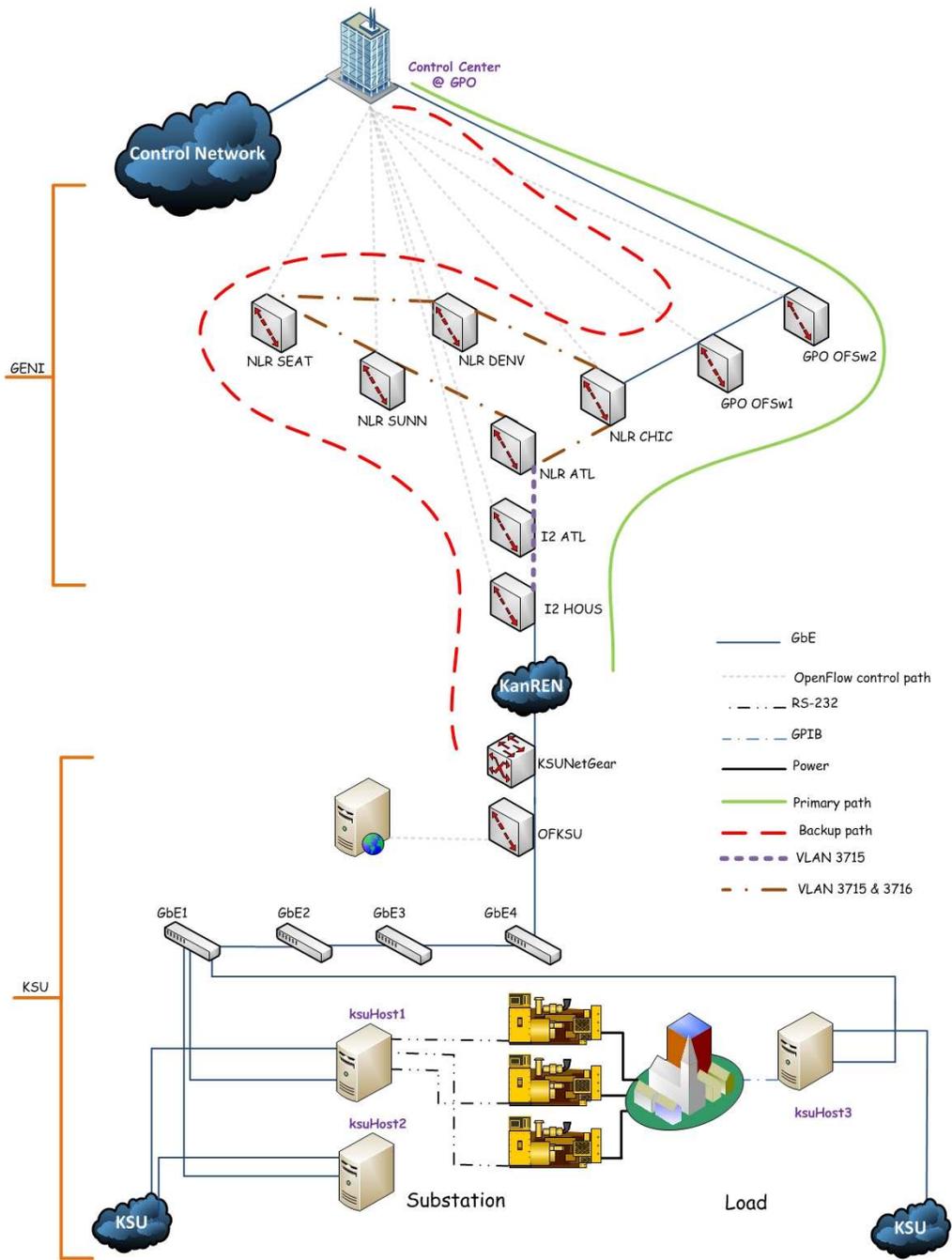


Figure 5.3: Integration of the power grid and communication network to realize a prototype of the smart grid. I2 and NLR correspond to OpenFlow switches in the research backbones of Internet2 and National LambdaRail. HOUS, ATL, SUNN, SEAT, DEV, and CHIC correspond to OpenFlow switches in Houston TX, Atlanta GA, Sunnyvale CA, Seattle WA, Denver CO, and Chicago IL, respectively.

5.2.3 OpenFlow Architecture

Figure 5.4 illustrates the fundamental components of the OpenFlow Architecture: flow table, secure channel, and OpenFlow protocol [65, 92, 66, 93]. As shown, the control and data planes are decoupled, a fundamental feature of Software Defined Networks.

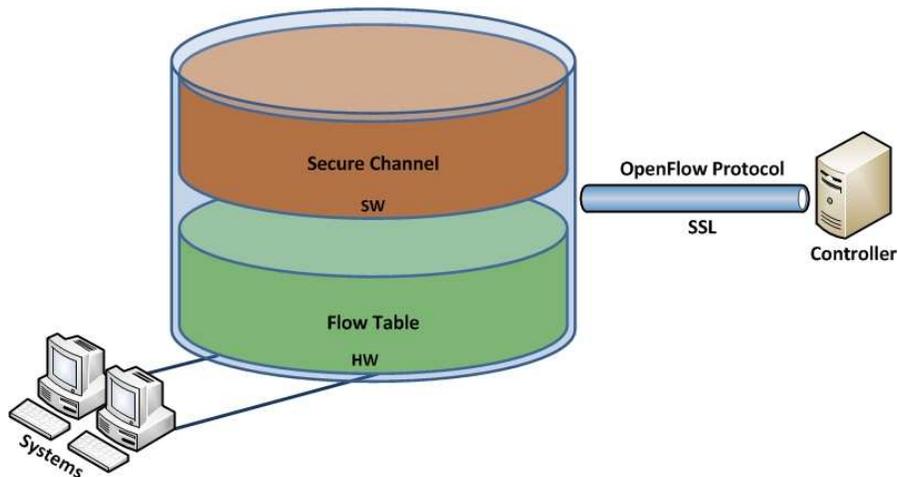


Figure 5.4: *OpenFlow architecture*

A high level description of the OpenFlow mechanisms is detailed below [65]. Note that a “flow” in OpenFlow is an abstract construct for a stream of packets with identical header fields. For example, there could be a TCP or UDP flow of packets.

1. The OpenFlow (OF) controller uses the OF protocol to install flow space rules in the flow table of the OF switch preemptively or at run time.
2. As flows arrive at the OF switch, they are checked against a list of flow space rules in the flow tables.
3. If a packet from a stream does not match any rule in the flow table, the first packet of this stream is encapsulated and transmitted to the OF controller as a “packet-in” message for further evaluation.

4. After evaluation, the OF controller installs a new rule for this type of packet. All subsequent packets encounter similar actions without visiting the OF controller.

5.3 Experiments

5.3.1 Software Defined Networking-Traffic Engineering (SDN-TE)

Traffic engineering can be defined as steering traffic in under utilized links. The MPLS solution involves the three phases: 1) creating tunnels, 2) routing traffic through these tunnels, and 3) using tunnel features for management. Phase one includes topology discovery using Interior Gateway Protocols (IGPs) such as Open Shortest Path First (OSPF), path calculation using protocols such as Constrained Shortest Path First (CSPF), and label distribution using protocols like Resource Reservation Protocol (RSVP). Phase two includes mechanisms for static routing, Policy Based Routing (PBR), and auto-route. Finally, phase three includes features such as auto-bandwidth and fast reroute for tunnel management. Our SDN-TE solution also has the three similar phases of 1) creating flows, 2) forward traffic through flows, and 3) using flow features for management. For phase one, we utilize the Link Layer Discovery Protocol (LLDP) as the foundation of the discovery OF application (to discover the topology). This application is integrated into our Core OF application that maintains a module which reads the flow description from an external file. Flows are described in a tuple that contains the following seven elements:

1. Source datapath/switch identification (dpid: in decimal)
2. Destination dpid (in decimal)
3. Flow priority: an integer ranging from 0-7 where 0 represents the most important flow
4. Reserved bandwidth: an integer representing the allocated bandwidth in Mbps
5. “Yes” or “no”: whether sub-pools will be implemented
6. Traffic type: such as User Datagram Protocol (UDP) or Transmission Control Protocol (TCP)

7. “Yes” or “no” (whether auto-bandwidth is enabled)

CSPF is also used for path calculation and, finally, OpenFlow is the protocol used to install flows. For phase two, our auto-routing module forwards traffic through flows, and phase three utilizes traffic engineering features similar to that of MPLS. To obtain the actual throughput measurements from the dpids, we integrated the default OpenFlow “Monitoring” application with our Core OF application and utilized the switch statistics application programming interface (API).

Prior to deploying the Demand Response (DR) application, we demonstrate the traffic engineering features of our OF controller on GENI. In Figure 5.5, the top graph captures traffic through the backup path and the bottom captures traffic through the primary path of Figure 5.3. The auto-route module initially installs four flows of UDP traffic with the respective port numbers 6000-6003. Load balancing is disabled and all flows are routed through the shortest path. Auto-bandwidth is enabled for flow 6003. The global reservable bandwidth for each link is set to 450Mbps and reserved bandwidth for flows 6000-6002 is 110Mbps, resulting in a total reserved bandwidth of 330Mbps. Using a shell script, the 6003 flow is incremented at pre-defined intervals. As shown in the bottom graph, the auto-bandwidth mechanism reflects the actual throughput of flow 6003. When the total capacity exceeds the global reservable link bandwidth (450Mbps), the preemption mechanism is activated and flow 6003 is rerouted through the long path.

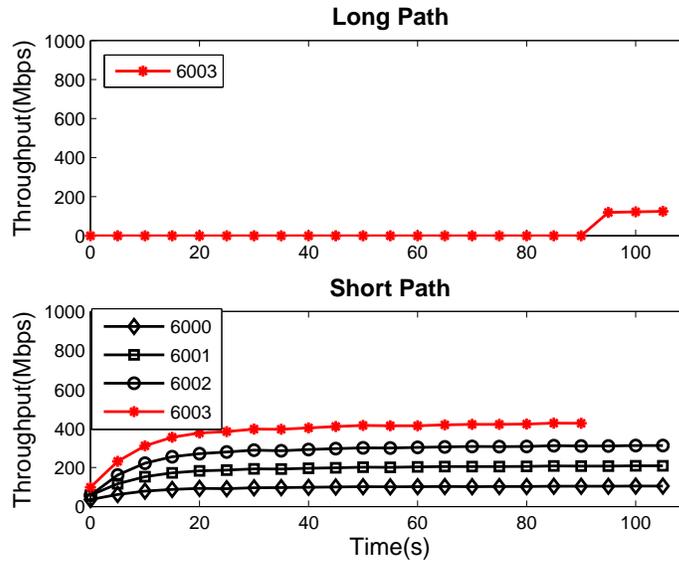


Figure 5.5: OpenFlow traffic engineering services on GENI

5.3.2 Demand Response

The following experiments considers a DR smart grid application which utilizes load shedding to regulate the power grid's frequency. More specifically, customers provide consent to utility companies seeking to regulate the on/off period of electric appliances to reduce the load during peak periods of demand. In exchange, customers receive fringe benefits such as a lower rate for electricity. What has this achieved? During peak periods, usually between the hours of 5pm to 7pm, residents return home from work and school, and air conditioning units, washers, dryers, and stoves are turned on. It is during this critical period of increasing load that utility companies have to choose to do nothing and risk cascading failures, "fire-up" backup generators, which could cost thousands of dollars and will be turned off at the end of the two-hour period, or seek alternative means to reduce the peak demand. Therefore, demand response is a compromise that reduces the demand and results in financial rewards for both the utility and the customer. However, demand response is as efficient and reliable as the supporting network infrastructure.

For all experiments, the objective is to maintain the nominal frequency of 60Hz. We consider three synchronous generators providing electricity to fixed and variable loads where the variable

loads represent appliances such as air conditioning units that can be toggled off and on. The Generator Agent (GA) at ksuHost1 transmits the period measurement corresponding to the analog frequency of the generators to the Control Center Agent (CCA). The CCA then executes Algorithm 4 as part of the load shedding control logic and transmits load shed measurements to the Load Agent (LA):

Algorithm 4 Control Logic for Load Shedding

f_{nom} :=Nominal frequency of 60Hz
 f_{act} :=Actual frequency from load agent
 f_{dev} :=The deviation of the actual frequency from the nominal frequency
 f_{thres} :=The frequency threshold was set to 0.1
 K_p :=The gain for the control system which was 5
 R_i :=Initial resistance of 200 Ω
 R_{sf} :=Resistance scaling factor
 R_{new} :=New resistance
for (;;) **do**
 Convert period (in seconds) from GA to frequency (in Hz)
 $f_{dev} = f_{nom} - f_{act}$
 if ($f_{dev} > f_{thres}$ or $f_{dev} < -f_{thres}$)
 $R_{sf} = f_{dev} K_p$
 $R_{new} = R_{sf} + R_i$
 Transmit R_{new} to the LA at ksuHost3
 $R_i = R_{new}$
 end if
end for

To trigger a deviation from the nominal frequency, we fail G3. As the frequency deviates from the nominal value of 60Hz, the CCA utilizes the logic in Algorithm 4 to transmit load adjustment measurements to the LA. The LA in turn adjusts the variable load accordingly to achieve the nominal frequency. We conduct this experiment under the following conditions:

1. With the CCA at KSU in order to obtain the benchmark frequency response and load shedding profile
2. During a failure on the primary path where fail-over mechanisms are implemented to reroute traffic onto the backup path

3. Injecting streams of traffic to congest the network
4. Streams of traffic are load balanced through the network

For all experiments, we compare the frequency response and load shed profile to that of the benchmark case where the CCA was deployed at KSU. Initially, we conducted the DR experiment without load control and observed that all generators quickly went outside their operating limits within 5s. The following subsections present details of the three experiments and the results obtained.

5.3.3 Experiment 1: Automatic Fail-over

To demonstrate the automatic fail-over mechanism, it was necessary to create a logical link failure on the primary path. To accomplish this task, we modified the discovery module of the Network Operating System (NOX) package that utilizes the Link Layer Discovery Protocol to establish the network topology. Algorithm 5 realizes a link failure for a given source-destination pair of adjacent OpenFlow switches:

For this experiment, we only considered protection traffic in the network (i.e. traffic between agents). At approximately 27s into the experiment, we failed G3 and, as shown in Figure 5.6, the frequency began to deviate from the nominal value. Approximately 29s, we also failed a link on the primary path. Figure 5.7 captures the throughput in the backbone network for the duration of the experiment. In particular, the automatic fail-over mechanism was able to reroute all traffic in less than 20s, considering a bi-directional distance of thousands of miles. Given a Round Trip Time (RTT) of 200ms (as opposed to .09ms for the benchmark case), Figure 5.8 shows an increase of 20 Ω of load shedding.

Algorithm 5 Link Failure Algorithm

$A_{i,j}$:=The adjacency structure that contains all source-destination (i, j) dpids (i.e. OpenFlow datapaths/Switches) and the time of discovery

t_f :=The time to fail a link

t_c :=The current time

t_l :=The time set to delete a link from A_{ij}

src :=The source dpid that connects the link to be removed

dst :=The destination dpid that connects the link to be removed

for (;;) **do**

 Update the topology using LLDP

if $i \in A_{ij} == src$ and $j \in A_{ij} == dst$ and $t_c > t_f$

 Do not update the time for link $A_{src,dst}$

else

 Update adjacent dpids in the A_{ij}

end if

if $t_{A_{i,j}} > t_l$

 Link (i, j) has timed out

 Delete i, j from $A_{i,j}$

end if

end for

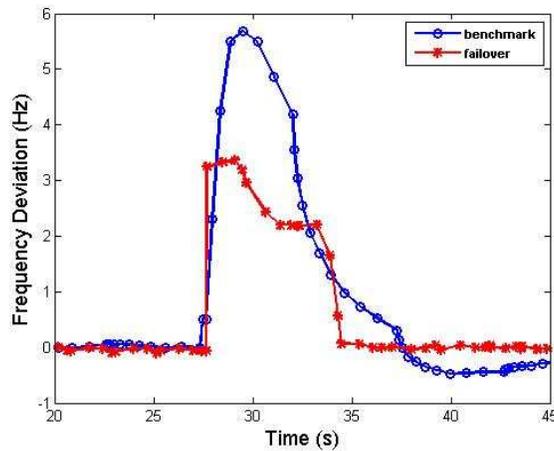


Figure 5.6: Comparison of the frequency response for a failure at G3 as an automatic fail-over mechanism reroutes traffic through the backup path to the benchmark experiment where CCA resided at KSU

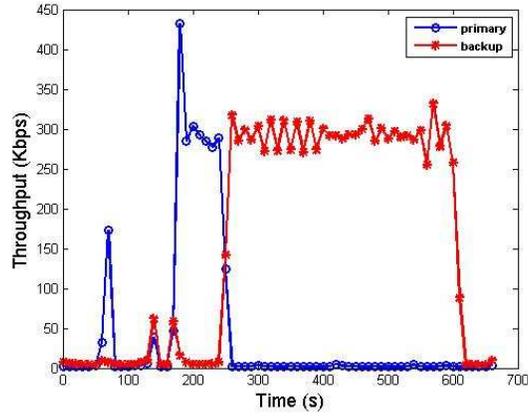


Figure 5.7: Throughput through the backbone network as automatic fail-over ensues

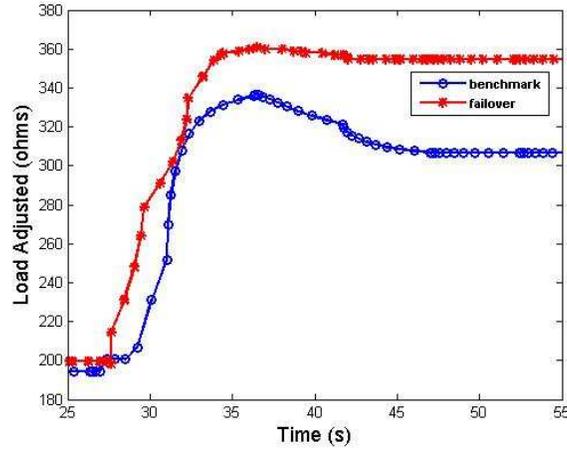


Figure 5.8: Comparison of the the load shed profile as an automatic fail-over mechanism reroutes traffic from the primary to the backup path

5.3.4 Experiment 2: Congestion

In this experiment, we investigated the impact of congestion on frequency response and load shedding profile. In particular, we used the queuing mechanism of the OpenFlow switch in KSU to transmit 5 TCP streams of 190Mbps and 1 TCP stream of 50Mbps with the objective of incrementally “filling the pipe” with 1Gbps (which is the capacity of the GENI backbone network) of “background” traffic. Figure 5.9 demonstrates the “max rate” queuing feature of the Pronto 3290 OpenFlow switch at KSU. Two 900Mbps streams originate from two source hosts, destined to

a single destination host. Within the first 6s to 14s, the congestion control mechanisms of TCP result in a throughput of approximately 50Mbps, as compared to the 900Mbps throughput realized by the User Datagram Protocol (UDP). At 15s, queues of 500Mbps are installed and, as shown, both TCP and UDP streams attain a throughput of 500Mbps. At 38s, the queues are removed and original behavior resumes.

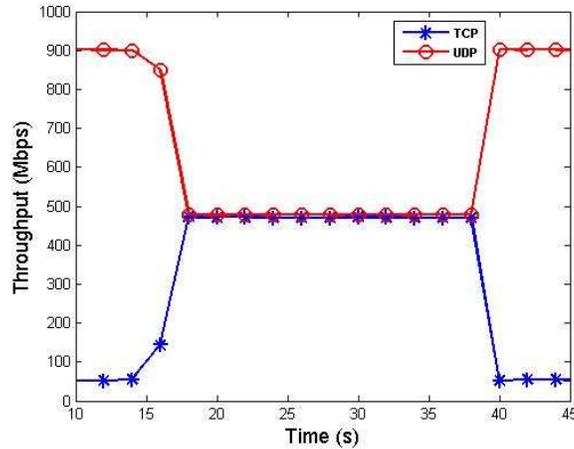


Figure 5.9: Max rate feature for queues on the Pronto 3290 OpenFlow switch

In this experiment, the streams originating from ksuHost2 are destined for the host at the Control Center. Figure 5.10 shows a cross section of the rate at which packets are transmitted by the GA and received by the CCA. The “generator” stream represents protection traffic transmitted by the GA and “ x streams” represent the number of streams x , that existed through the backbone network on the primary path. As shown, though the number of streams increase, arrival rate at the CCA remains fairly constant (1 packet every second) with a slight delay between the transmission and arrival of packets. This is expected as the RTT was 98ms on the primary path. Initially, protection packets were transmitted every 200ms; however, the generators quickly surpassed their operating limits since the rate at which the frequency measures are updated at the CCA far exceeded the rate at which load shed commands were executed by the variable load. This was not expected, as the RTT was 98ms. However, considerable delays exist on the path from the LA through the GPIB connection to the variable load. Perhaps a more efficient variable load would

resolve this issue.

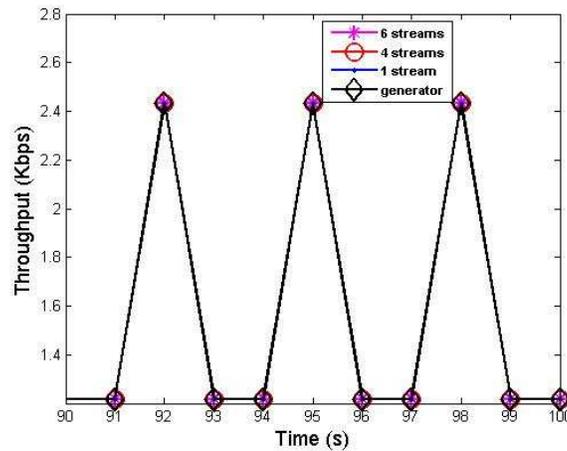


Figure 5.10: Rate at which packets are received at the Control Center from the GA

From Figure 5.11, it appears that throughput times overlap for the various streams. However, Experiment 2 consists of four individual trials and the different throughput values have been aggregated into one plot where the start time of 0s signifies the initial injection of a given number of streams into the network. Specifically, for each trial, we transmitted x stream/s through the network, failed G3 and recorded the frequency response and load shed profile for this trial.

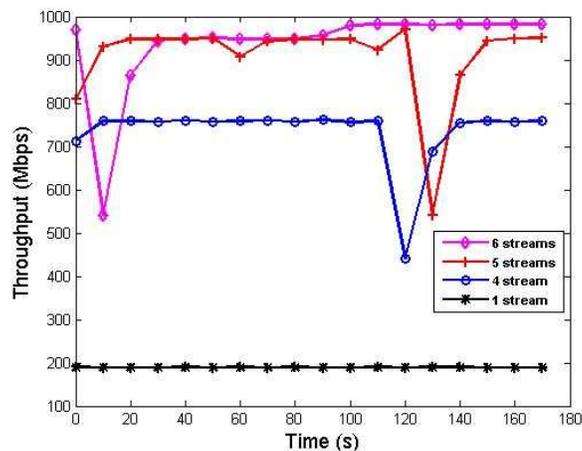


Figure 5.11: Throughput in the primary path as streams are incrementally traversing the network

From Figure 5.12, the settling time (i.e. the difference between the times when the frequency

deviates and when the frequency returns to the nominal value) for all streams is approximately 10s. More importantly, though the frequency returns to the nominal value at 8s for the benchmark experiment, the settling time was 20s. This can be attributed to the high gain used to compensate for delays between transmission of a protection packet from the GA to the CCA, and the execution of a load shed command from the CCA to the LA. This high gain substantially increases the step size of the resistance measures transmitted to the load. As shown from the benchmark plot, a high gain and small delay results in an “overshoot” of the ideal resistance value necessary for the frequency to return to the nominal value. A smaller gain would result in a graceful return of the frequency to the nominal value for the benchmark frequency. However, this low gain would increase the settling time for experiments with high latencies between GA, CCA, and LA.

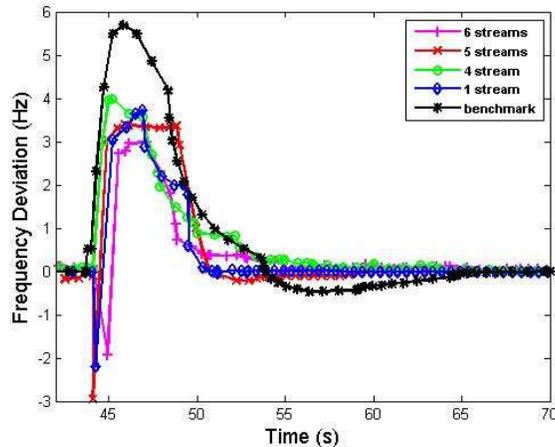


Figure 5.12: Comparison of the frequency responses for each trial where x stream/s traversed the network in addition to protection traffic from LA

Figure 5.13 shows a range of 80 ohms for all streams. This can be attributed to the absence of queuing mechanisms on the return path from the CCA to the LA, in addition to inaccuracies in the initial configuration of the power system.

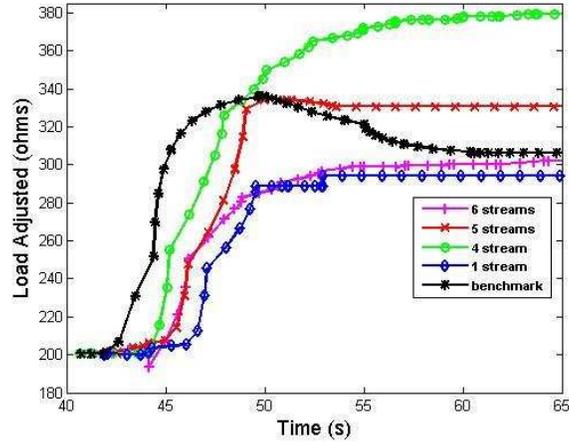


Figure 5.13: Rate at which packets are received at the Control Center from the LA

5.3.5 Experiment 3: Load Balancing

For this experiment, we first load balanced three streams on the primary path and three streams on the backup path, and then executed the failure of G3. We then compared the result to a second experiment where we injected all six streams in the primary path and created a separate queue of 10Mbps for the protection traffic. Figure 5.14 displays the throughput in both paths as streams are load balanced. As expected, the streams through the backup path realized an individual throughput of approximately 80Mbps (as opposed to about 200Mbps in the primary path) due to the 200ms latency. This is a direct result of the flow control and congestion mechanisms inherent within the TCP (Transmission Control Protocol) Iperf streams where long latencies exist.

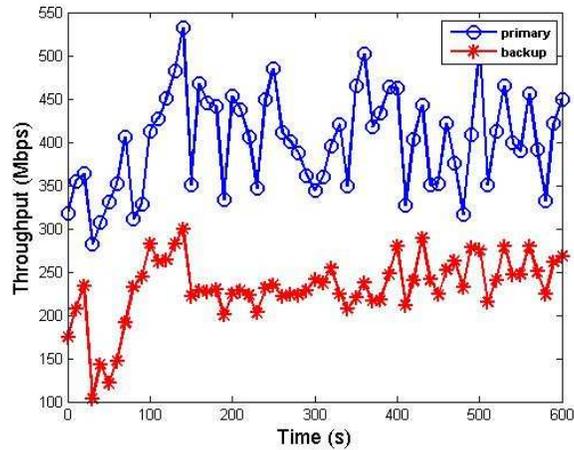


Figure 5.14: *Throughput on both paths as streams are load balanced in the backbone core network*

From Figures 5.15 and 5.16, the results from both QoS and load balancing were similar to that of the benchmark case. In particular, the QoS experiment shows a graceful return to the nominal frequency. From the load balancing result, the frequency response and load shed profile is similar to that of the benchmark case where the resistance necessary to allow the frequency to return to the nominal value is exceeded. This may be attributed to the sequence in which protection and background packets are transmitted through both paths, in addition to the sequence in which they arrive at the Control Center.

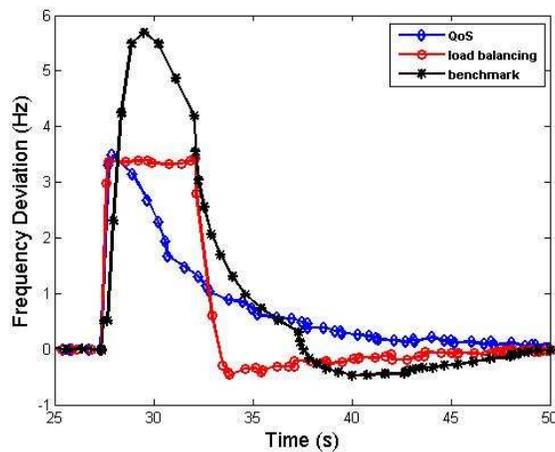


Figure 5.15: *Comparison of frequency response for the QoS, load balancing and benchmark experiments*

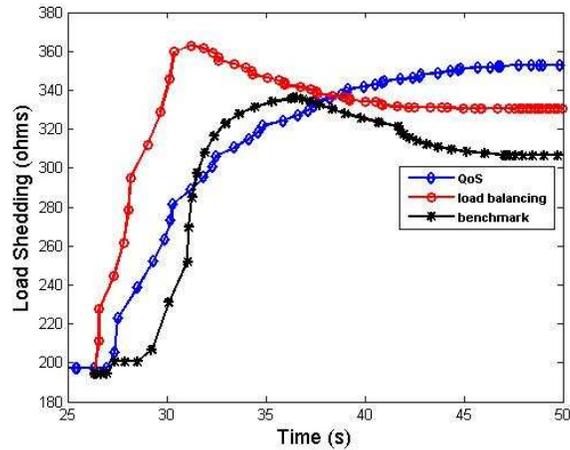


Figure 5.16: Comparison of the load shed profile for the QoS, load balancing and benchmark experiment

5.3.6 Fast Reroute (SDN-TE Protection)

We create a separate section for this experiment as the testbed used was located in the Smart-Grid lab at K-State. Here we compare the Fast Reroute (FRR) mechanism of MPLS to OpenFlow using hybrid routers that support both protocols. Figures 5.17 and 5.18 illustrate the network configuration for MPLS and OpenFlow, respectively. In particular, the source host (Src) transmits traffic to the destination host (Dst). The primary traffic routes are maintained by Sw1, Sw2, and Sw4 and the backup routes by Sw1, Sw3, and Sw4. We insert a Gigabit Ethernet (GbE) switch for the purpose of ensuring fairness in our comparison such that the routers are not aware of any physical disconnections (which is usually the case in real-world scenarios where network media is cut). Therefore, to execute a link failure, we disconnect the link from the GbE switch adjacent to Sw1.

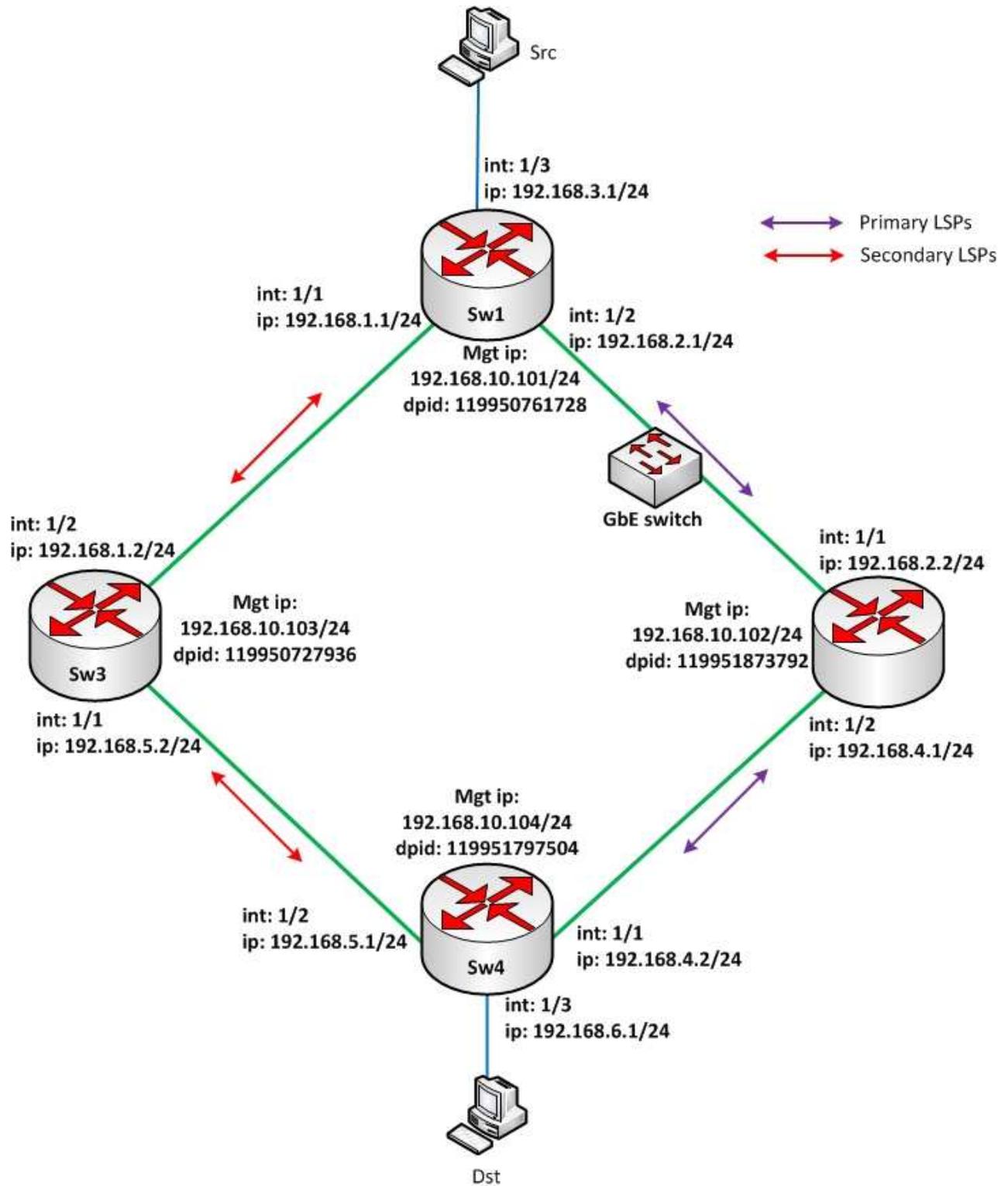


Figure 5.17: MPLS network configuration

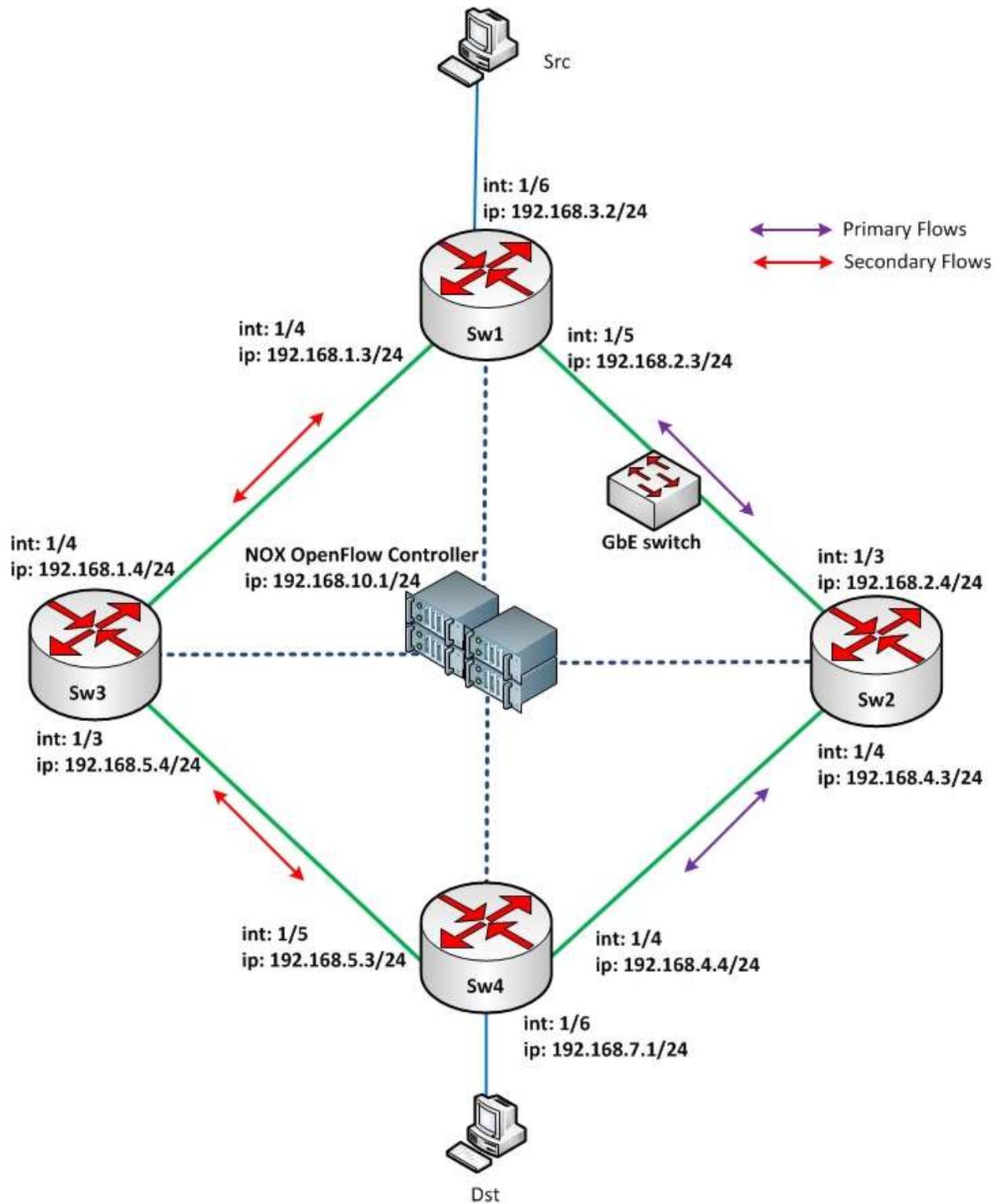


Figure 5.18: *OpenFlow network configuration*

During our initial tests with the MPLS FRR mechanism, we observed that when we reconnected the link after a link failure, the primary LSP encountered issues while being reconstructed. For this reason, we used the hot-standby mode of MPLS as opposed to the FRR mode. It is worth mentioning that for our experiments, the operation of both modes are identical. In particular, in FRR mode, RSVP is used to establish multiple paths for LSPs and the resources for the primary as well as the secondary path are allocated. Furthermore, FRR has preemption capabilities which allow an alternate path to be utilized if one exists that is more attractive than those previously allocated. With hot-standby, resources for both primary and secondary LSPs are allocated. However, the allocation is manual. Unlike the flexibility of FRR, even though more attractive paths exist, you are restricted to the defined paths.

For our first experiment, we generate Internet Control Message Protocol (ICMP) pings from Src to Dst. For both protocols, we first begin by transmitting pings at 1ms intervals, then fail the link between Sw1 and Sw2. We then repeat this experiment ten times and record the maximum number of packets dropped. This reflects a worse-case scenario. We then increment the ping interval to 5ms and repeat the same procedure up to a 50ms ping interval. Figure 5.19 shows that our OpenFlow implementation drops less packets than its MPLS counterpart.

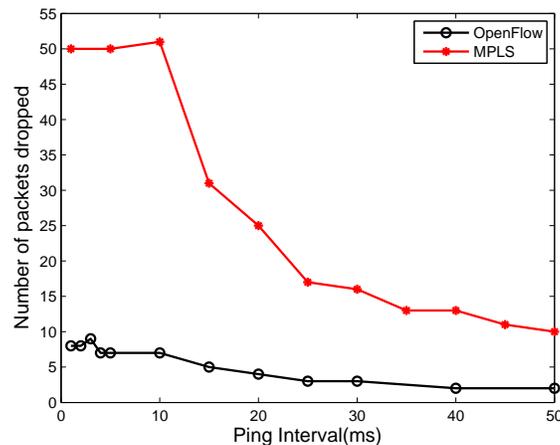


Figure 5.19: Comparison of OpenFlow and MPLS FRR mechanisms as ICMP ping packets are transmitted

For our final experiment, we use the Iperf tool to generate TCP traffic from Src to Dst. While

traffic is being generated, we execute a link failure and capture the drop in throughput. As shown in Figure 5.20, the drop in throughput for the OpenFlow implementation is less than that of the MPLS equivalent.

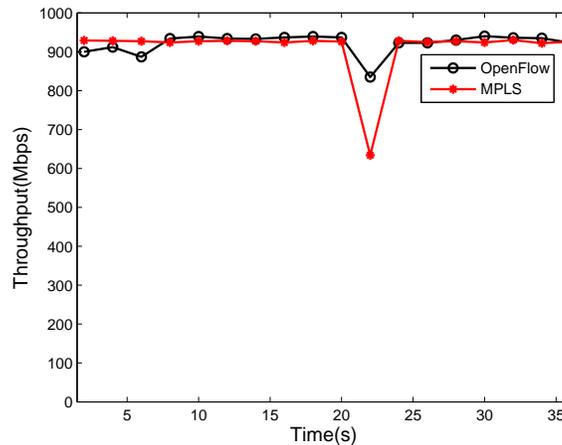


Figure 5.20: Comparison of OpenFlow and MPLS FRR mechanisms as Iperf TCP packets are transmitted

5.4 Discussion

This project is a first-cut exploration into the current capabilities of hardware that supports the OpenFlow technology for smart grid operations. In particular, we investigated whether OpenFlow could provide an automatic fail-over mechanism and traffic engineering services such as auto-route, auto-bandwidth, and fast reroute, using a controller developed within a two-week period. Other traffic tunneling mechanisms were implemented in the simulative environment of mininet. However, during the actual deployment process we learned that the HP, NEC, and Pronto switches within GENI do not support a unified set of actions at all layers in the hardware path, thus hindering any attempt to rewrite a packet for tunneling purposes. Furthermore, an attempt to utilize the exploratory pool of the 6-bit Differentiated Service Code Point (DSCP) field for tagging packets proved unsuccessful, as these values were translated into a level of service in the network and generally resulted in increased RTTs of up to 1s.

For all GENI experiments, we used OpenFlow to create queues for all traffic streams on egress

at KSU. Since the initial aggregated throughput on both primary and backup paths was approximately 10Kbps, we determined that the sequencing of packets at the queues would be unchanged as they traversed the backbone network. For this reason and in addition to the flow control and congestion control mechanisms of TCP, all frequency response profiles were identical when all streams traversed the primary path. However, no queues were created for traffic from the CCA to the LA. This contributed to variations in load shed profiles for different experiments. Assuming a high volume of traffic on the backbone GENI network, it would be necessary to implement queues on other switches on the path to provide similar results.

From a power grid perspective, Control Centers and substations are generally in proximity to each other (as opposed to spanning multiple states within the US, as was done in this project). Furthermore, specialized mechanisms are incorporated into substations to provide a more accurate reading of the generator frequencies. For this reason, a frequency deviation greater than 0.5 would cause generators to go offline [76]. The mechanisms used in this project consisted of off-the-shelf and in-lab components. Therefore, this work was a “proof-of-concept” that current hardware can be used to implement the afore mentioned features.

5.5 Summary

Previous research shows that current software switches can be used to provide MPLS features using an OpenFlow control plane. To date, the available hardware does not readily support MPLS. However, can we use OpenFlow with the commercially-available hardware in GENI to provide MPLS-like features? Given a short period of two weeks and the limitations of the current hardware, we implemented and deployed an OpenFlow controller that provided automatic fail-over mechanisms and traffic engineering services. These services were used to support real traffic from cyber physical systems in a smart grid Demand Response experiment that utilizes load shedding to regulate frequency. Finally, we constructed a test in the K-State Smart-Grid lab to compare the fast reroute mechanism of MPLS to that of our OpenFlow application. For the given network scale, the results demonstrate that OpenFlow can have a higher performance level than MPLS.

This work demonstrates the flexibility and speedy implementation and deployment of a real-world solution under real-world network conditions. Within a short period of time, we were able to run complex experiments that span resources in multiple spatial locations from Kansas to Texas to locations on the West Coast such as California and Washington, and Boston on the East Coast. It goes without saying that deploying an experiment of such magnitude on GENI is rather complex and requires a learning curve of the various tools and mechanisms available. Furthermore, there exists a lag in the current capabilities of the hardware when compared to the capabilities defined in the OpenFlow specification.

Chapter 6 concludes the dissertation by discussing the applicability and benefits of this work in evaluating software defined networks for communication and control of cyber physical systems.

Chapter 6

Conclusions and Future Research

6.0.1 Review

This dissertation evaluates software defined networking for communication and control of cyber physical systems where the system under consideration is the smart grid. Before evaluating SDN for the smart grid, we first highlight the importance of designing a robust physical infrastructure as this sets the foundation for obtaining an even higher level of performance when combined with software mechanisms.

To date, robustness in complex networks is an ongoing research effort. Among other topological measures, we use algebraic connectivity from spectral graph theory as our measure of robustness; the larger the algebraic connectivity, the more robust the network. Since we know where to add a link to maximally increase algebraic connectivity, we extend this idea to answer the question of “Where should an edge be rewired to increase algebraic connectivity the most?” From a panoramic perspective, if we can show that rewiring links yields the same robustness as adding links, one would opt for a rewiring solution since the cost to constantly add links can become prohibitive. From our analytical results, we conclude that the greatest increase in algebraic connectivity tends to occur when we disconnect a link between two strongly connected vertices and attach a link between two weakly connected vertices. To validate these results, we apply the rewiring strategy on three classes of networks.

From our simulations, we initially compare graphs from the three classes of networks to determine the class that has the highest increase in algebraic connectivity. Our results reveal that

graphs from Gilbert's model (Gi) tend to have the lowest initial value for algebraic connectivity in addition to the highest increase in algebraic connectivity after rewiring. For all classes, we then compared the increase in algebraic connectivity achieved by rewiring as opposed to adding links. When we rewired or added links in excess of 5% of the total set of links, the value of algebraic connectivity for adding links increased monotonically and was greater than the value for rewiring. However, the number of links rewired or added in the real-world networks is relatively small due to a high cost factor. In such cases, the increase in algebraic connectivity is similar for both rewiring and adding links. Therefore, one can conclude that a solution that rewires edges is as robust as a solution where edges are added. Finally, unlike the monotonic increase in algebraic connectivity observed when links are added, there exists a rewiring threshold which, once surpassed, algebraic connectivity remains constant.

At this point, we have developed methods that maximally increase the algebraic connectivity of a network and hence, increase its robustness. However, what is the impact of increasing algebraic connectivity in real-world networks? What is the impact when we increase the robustness of a network? In response to these questions, we created a communication network identical to a power grid network. We then rewired and added links to this original network to create alternate variations that improve on the algebraic connectivity of the original. Finally, we created two scenarios where we first evaluated the topological impact on increasing algebraic connectivity for each network. For the second scenario, we inserted each network in a hybrid simulator to evaluate the impact to traffic characteristics as algebraic connectivity is increased.

From our topological results, we deduce that adding/rewiring links creates a more homogeneous network with regards to the node degree distribution. Our hybrid simulations revealed an increase in network performance. In particular, increasing algebraic connectivity generally reduced PLR, RTT, and increased the throughput of a network. This implies that a network that may be ideal for the power grid, may not necessarily be ideal for the communication network. Furthermore, there were instances where rewiring a network resulted in the same performance values as adding links. For utility companies at the design phase of deploying a communication

infrastructure, rewiring a network to increase algebraic connectivity may be more cost-effective than adding links.

At this point, we delved into evaluating SDN for communication and control of smart grids. To this end, we were concerned with comparing the “state-of-the-art” networking technology, MPLS, to OpenFlow. As we mention in the future work, it will be interesting to build on the previous work of increasing algebraic connectivity to evaluate the performance increase in both power system and communication network.

This work is the initial step towards demonstrating that an inexpensive SDN OpenFlow technology can perform as well as MPLS for transmission operations in the smart grid. Specifically, we integrate an AC power systems simulator with a communication network simulator to realize the functionalities of the smart grid. We first compared the performance of the two networking technologies by considering measures such as PLR, RTT, and throughput. Secondly, we compared the performance on the power system as each technology was employed by evaluating voltage and frequency profiles. The results indicate that configuring the OpenFlow network similar to that of an MPLS network provides similar performance levels to MPLS. This configuration includes the preemptive installation of flows such that the timeout parameter of OpenFlow exceeds the completion of transmissions.

We have demonstrated via simulation that OpenFlow can perform as well as MPLS for smart grid transmission operations. Furthermore, researchers have used software switches to show that OpenFlow can provide similar features to MPLS. To date, the available hardware does not readily support MPLS. For this reason, this work demonstrates how the flexibility and programmability of OpenFlow can be used in commercially-available hardware in GENI to provide MPLS-like features. In particular, we created a prototype of the smart grid using power system components at K-State and communication networking components of GENI, similar to the functionality of the hybrid simulator previously introduced. We implemented and deployed an OpenFlow controller that provided traffic engineering services identical to that of MPLS but using the OpenFlow control plane. This work demonstrates the flexibility and speedy implementation and deployment of a

real-world solution under real-world network conditions. Within a short period of time, we were able to run complex experiments that span resources in multiple spatial locations. Furthermore, we were able to compare the fast reroute capabilities of MPLS to that of OpenFlow.

6.0.2 Future Work

From the topological component of this dissertation, it will be interesting to consider the impact to a network's characteristics when algebraic connectivity is maximally increased. Such networks can include complex networks such as communication, power grid, and transportation networks. It would also be interesting to consider rewiring edges to maximally increase other spectral measures such the spectral radius of a network. Finally, the "greedy" algorithm employed in this chapter will not necessarily result in the optimal increase in algebraic connectivity. As a result, it would be interesting to explore various strategies to optimize algebraic connectivity when multiple links are rewired.

To evaluate the impact of increasing algebraic connectivity in real-world networks, all links were given a uniform weight of 1. The future work includes developing a mathematical variant of algebraic connectivity that considers both topology and traffic.

To compare the performance of MPLS and OpenFlow, we created a hybrid simulator that integrated the continuous-time behavior of the power system with the discrete-event behavior of the communication network. Both protection and background traffic were transmitted between the control center and substations as the performance of both the communication network and power system were evaluated. Future work should quantify the protection traffic arriving at the Control Center and investigate protection schemes in both MPLS and OpenFlow. Furthermore, experiments should be developed using realistic smart grid traffic profiles for the communication network and deploying OpenFlow in real-scale, real-time, and on a real-world testbed, such as GENI, that conforms to and exceeds the current QoS and security standards established by entities such as NERC.

For our smart grid prototype, we utilized power system components at K-State and commu-

nication network components of GENI to evaluate SDN for communication and control of transmission operations. The current hardware limited the granularity of defining flows such that flows were defined using headers of Layer 2 to Layer 4. As this obviously does not scale, it would be interesting to run similar experiments such that flows are as flexible as that provided in the OpenFlow specification. In addition, as the GENI testbed expands, it would be interesting to evaluate the increase in performance of both the communication network and the power system as algebraic connectivity is maximized. From the power system's domain, it will be of interest to consider an algorithm that dynamically selects an optimal gain measure given a latency measure between the generator agent (GA), control center agent (CCA), and load agent (LA). Finally, since fast reroute is crucial to any cyber physical system, it would be interesting to directly compare MPLS to OpenFlow on GENI.

Bibliography

- [1] S. Abraham and R. J. Efford. US-Canada Power System Outage Task Force. [online]. Available: <https://reports.energy.gov/>, 2004.
- [2] R. W. Cummings and E. H. Allen. August 14 Blackout and Subsequent Investigation, 2003.
- [3] R. Bacher, U. Naf, M. Renggli, W. Buhlmann, and H. G. Nussbaumen. Report on the Blackout in Italy on September 28, 2003.
- [4] D. W. Hilt. Impacts and Actions Resulting from the August 14 blackout, 2006.
- [5] P. McCurley, V. Whitaker, S. Bacik, C. Kotting, P. Myrda, T. Siegfried, and M. Ilic. Reliability Considerations from the Integration of Smart Grid. [online]. Available: <http://www.nerc.com>, 2010.
- [6] F. Sissine, “Energy independence and security act of 2007: A summary of major provisions,” in *CRS Report for Congress RL34294*, 2008.
- [7] S. Shengnan, M. Pipattanasomporn, and S. Rahman, “Challenges of phev penetration to the residential distribution network,” in *Power and Energy Society General Meeting*, pp. 1–8, 2009.
- [8] D. E. Bakken, A. Bose, C. H. Hauser, D. E. Whitehead, and G. C. Zweigle, “Smart generation and transmission with coherent, real-time data,” in *Proceedings of the IEEE*, vol. 99, pp. 928–951, 2011.
- [9] D. E. Bakken, C. H. Hauser, H. Gjermundrod, and A. Bose, “Towards more flexible and robust data delivery for monitoring and control of the electric power grid,” in *Technical report EECS-GS-009*, 2007.

- [10] R. H. McClanahan. [Online]: <http://www.nerc.com/>, 2003.
- [11] A. Akyamac, J. Deshpande, A. McGee, CISSP, GREM, and GCIH, “Achieving nerc cip compliance with secure mpls networks,” in *Technology White Paper*, pp. 1–28, 2010.
- [12] T. Hulsebosch, D. Belmont, and M. Manske, “Smart grid network: Mpls design approach,” 2011.
- [13] S. Das, “Unified control architecture for packet and circuit network conformance,” in *PhD Thesis*, 2012.
- [14] S. Whyte and Google, “An open-source interoperable mpls lsr,” in *NANOG*, 2010.
- [15] N. McKeown, “How should the internet evolve?,” in *Kailath Symposium, Stanford University*, 2010.
- [16] D. Alderson, L. Li, and C. D. W. Willinger, “Understanding internet topology: Principles, models, and validation,” in *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 13, pp. 1205–1218, 2005.
- [17] J. C. Doyle, D. L. Alderson, L. Li, S. Low, M. Roughan, S. Shalunov, R. Tanaka, and W. Willinger, “The robust yet fragile nature of the internet,” in *PNAS*, vol. 102, pp. 14497–14502, 2005.
- [18] A. Sydney, C. Scoglio, P. Schumm, and R. Kooij, “Elasticity: Topological characterization of robustness in complex networks,” in *IEEE/ACM Bionetics*, 2008.
- [19] A. Sydney, C. Scoglio, M. Youssef, and P. Schumm, “Characterizing the robustness of complex networks,” in *International Journal of Internet Technology and Secured Transactions*, vol. 2, pp. 291–320, 2010.
- [20] C. Mass, “Transportation in graphs and the admittance spectrum,” in *Discrete Applied Mathematics*, vol. 16, pp. 31–39, 1987.

- [21] H. Wang and P. V. Mieghem, “Algebraic connectivity optimization via link addition,” in *Proceedings of IEEE/ACM Bionetics*, 2008.
- [22] D. Mosk-Aoyama, “Maximum algebraic connectivity augmentation is np-hard,” *Operations Research Letters*, vol. 36, no. 6, pp. 677 – 679, 2008.
- [23] H. Frank and I. Frisch, “Analysis and design of survivable networks,” in *IEEE Transactions on Communications Technology COM-18*, p. 567, 1970.
- [24] D. Bauer, F. Boesch, C. Suffel, and R. Tindell, “The theory and application of graphs,” pp. 89–98, 1981.
- [25] F. Harary, “On conditional edge-connectivity of graphs,” in *Networks*, vol. 13, p. 346, 1981.
- [26] A. H. Esfahanian and S. Hakimi, “On computing a conditional edge-connectivity of a graph,” in *Journal of Information processing Letters*, vol. 27, p. 195, 1988.
- [27] M. Krishnamoorth and B. Krishnamirthy, “Fault diameter of interconnection networks,” in *Computers and Mathematics with Applications*, vol. 13, p. 577, 1987.
- [28] V. Chv’atal, “Tough graphs and hamiltonian circuits,” in *Discrete Math*, vol. 5, p. 215, 1973.
- [29] H. Jung, “On a class of posets and the corresponding comparability graphs,” in *Journal of Combinatorial Theory B*, vol. 24, p. 125, 1978.
- [30] M. Cozzen, D. Moazzami, and S. Stueckle, “Seventh international conference on the theory and applications of graphs,” pp. 1111–1122, 1995.
- [31] M. Alon, “Eigenvalues and expanders,” in *Combinatorica*, vol. 6, p. 83, 1986.
- [32] B. Mohar, “Isoperimetric numbers of graphs,” in *Journal of Combinatorial Theory Series B*, vol. 47, p. 274, 1989.
- [33] R. Albert and H. J. A.-L. Barabasi, “Error and attack tolerance of complex networks,” in *Nature*, vol. 406, pp. 378–382, 2000.

- [34] A. Barrat, M. Barthelemy, and A. Vespignani, *Dynamical Process on Complex Networks*. Cambridge University Press, 2008.
- [35] S. Pahwa, A. Hodges, C. Scoglio, and S. Woods, “Topological analysis of the power grid and mitigation strategies against cascading failures,” in *Systems Conference*, pp. 272–276, 2010.
- [36] P. Mahadevan, D. Krioukov, M. Fomenkov, and B. Hauffaker, “Lessons from three views of the internet topology: Technical report,” in *Cooperative Association for Internet Data Analysis (CAIDA)*, 2005.
- [37] P. Mahadevan, D. Krioukov, K. Fall, and A. Vahdat, “Systematic topology analysis and generation using degree correlations,” in *ACM/SIGCOMM*, vol. 47, p. 274, 2006.
- [38] J. Dong and S. Horvath, “Understanding network concepts in modules,” in *BMC Systems Biology*, vol. 1, 2007.
- [39] D. L. Alderson, “Catching the network science bug: Insight and opportunity for the researcher,” in *INFORMS*, vol. 56, pp. 1047–1065, 2008.
- [40] S. Liu, X. Li, W. Jiang, and Y.-Z. Fan, “Adaptive synchronization in complex dynamical networks with coupling delays for general graphs,” in *Applied Mathematics and Computation, Elsevier*, vol. 219, pp. 83–87, 2012.
- [41] B. Mohar, “Eigenvalues, diameter, and mean distance in graphs,” in *Graphs and Combinatorics*, vol. 7, pp. 53–64, 1991.
- [42] X. Zhang, “Ordering trees with algebraic connectivity and diameter,” in *Linear Algebra and its Applications, Elsevier*, vol. 427, pp. 301–312, 2007.
- [43] A. Jamakovic and S. Uhlig, “Influence of the network structure on robustness,” in *Networks, 2007. ICON 2007. 15th IEEE International Conference*, pp. 278–283, 2007.
- [44] P. V. Mieghem, *Graph Spectra for Complex Networks*. Cambridge University Press, 2010.

- [45] Y. Fan, “On spectral integral variations of graphs,” in *Linear and Multilinear Algebra*, pp. 133–142, 2002.
- [46] S. Barik and S. Pati, “On algebraic connectivity and spectral integral variations of graphs,” *Linear Algebra and its Applications*, vol. 397, no. 0, pp. 209 – 222, 2005.
- [47] K. C. Das, “The laplacian spectrum of a graph,” in *Computers and Mathematics with Applications*, vol. 48, pp. 715–724, 2004.
- [48] A. Weinstein and W. Stenger, *Methods of Intermediate Problems for Eigenvalues*, vol. 89. Academic Press New York and London, 1972.
- [49] J. Batson, D. A. Spielman, and N. Srivastava, “Twice-ramanujan sparsifiers,” *Computer*, pp. 1–21, 2009.
- [50] D. Watts and S. Strogatz, “Collective dynamics of ‘small-world’ networks,” in *Nature*, vol. 393, pp. 440–442, 1998.
- [51] NWBTEAM. Network Workbench Tool, Indiana University, Northeastern University, and University of Michigan. <http://nwb.slis.indiana.edu>, 2008.
- [52] A.-L. Barabasi and E. Bonabeau, “Scale-free networks,” in *Scientific American*, vol. 288, pp. 60–69, 2003.
- [53] J. Nutaro. THYME. [Online]: <http://www.ornl.gov/1qn/thyme/docs/index.html>, 2010.
- [54] J. Nutaro and V. Protopopescu, “Calculating frequency at loads in simulations of electro-mechanical transients,” *IEEE Transactions on Smart Grid*, vol. 3, pp. 233–240, March 2012.
- [55] J. Arrillaga and N. R. Watson, *Computer Modelling of Electrical Power Systems, Second Edition*. Wiley, 2001.
- [56] R. Natarajan, *Computer-Aided Power System Analysis*. New York, New York: Marcel Dekker, Inc., 2002.

- [57] J. J. Nutaro, *Building Software for Simulation*. John Wiley and Sons, Inc., 2011.
- [58] Power Systems Test Case Archive [Online]:<http://www.ee.washington.edu/research/pstca/>.
- [59] L. Wenpeng, D. Sharp, and S. Lancashire, “Smart grid communication network capacity planning for power utilities,” in *Transmission and Distribution Conference and Exposition, 2010 IEEE PES*, pp. 1–4, 2010.
- [60] E. O. Schweitzer and D. E. Whitehead, “Real-time power system control using synchrophasors,” in *Protective Relay Engineers, 2008 61st Annual Conference for*, pp. 78–88, 2008.
- [61] Y. Ming-Jiang, T. Y. Jo, W. Dajun, and H. G. Kwong, “Cost effective ip camera for video surveillance,” in *Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on*, pp. 2432–2435, 2009.
- [62] X. Yang, “Designing traffic profiles for bursty internet traffic,” in *Global Telecommunication Conference, 2002. GLOBECOM '02.IEEE*, vol. 3, pp. 2149–2154, 2002.
- [63] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, “On the self-similar nature of ethernet traffic,” in *Networking, IEEE/ACM Transactions on*, vol. 2, pp. 1–15, 1994.
- [64] A. Aggarwal, S. Kunta, and P. K. Verma, “A proposed communications infrastructure for the smart grid,” in *Innovative Smart Grid Technologies, ISGT*, pp. 1–5, 2010.
- [65] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: Enabling innovation in campus networks,” in *SIGCOMM CCR*, vol. 38, 2008.
- [66] S. Das, G. Parulkar, and N. McKeown, “Unifying packet and circuit switch networks,” in *GLOBECOM Workshops*, pp. 1–6, 2009.
- [67] A. Tootoonchian and Y. Ganjali, “A distributed control plane for openflow networks,” in *Proceedings of NSDI Internet Network Management Workshop*, 2010.

- [68] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable flow-based networking with difane," in *ACM SIGCOMM*, 2010.
- [69] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. T. Gia, "Modeling and performance evaluation of an openflow architecture," in *23rd International Teletraffic Congress*, 2011.
- [70] A. Kessler, L. K. Skorin, O. Dobrijevic, M. Matijasevic, and P. Dely, "Towards qoe-driven multimedia service negotiation and path optimization with software defined networking," in *In Proceedings of the International Conference on Software, Telecommunications and Computer Networks*, 2012.
- [71] P. Fonseca, R. Bennesby, E. Mota, and A. Passito, "A replication component for resilient openflow-based networking," in *Network Operations and Management Symposium*, 2012.
- [72] H. E. Eglimez, B. Gorkemli, A. M. Tekalp, and S. Civanlar, "Scalable video streaming over openflow networks: An optimization framework for qos routing," in *Proc. IEEE International Conference on Image Processing*, 2011.
- [73] P. Dely, A. Kessler, and N. Bayer, "Openflow wireless mesh networks," in *IEEE International Workshop on Wireless Mesh and Ad Hoc Networks*, 2011.
- [74] R. Wang, D. Butnariu, and J. Rexford, "Openflow-based server load balancing gone wild," in *Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, 2011.
- [75] R. S. Braga, E. Mota, and A. Passito, "Lightweight ddos flooding attack detection using nox/openflow," in *In: 35th Annual IEEE Conference on Local Computer Networks*, 2010.
- [76] S. Hilliday. Report of the investigation into the automatic demand disconnection following multiple generation losses and the demand control response that occurred in May 2008. [Online]<http://www.nationalgrid.com>, 2008.

- [77] C. M. Davis, J. E. Tate, J. Okhravi, C. Grier, T. J. Overbye, and D. Nicol, "Scada cyber security testbed development," in *Power Symposium, 2006. NAPS 2006. 38th North American*, pp. 483–488, 2006.
- [78] N. Higgins, V. Vyatkin, N. K. C. Nair, and K. Schwarz, "Distributed power system automation with iec 61850, iec 61499, and intelligent control," in *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, pp. 81–92, 2011.
- [79] J. N. Paquin, J. Moyon, G. Dumur, and V. Lapointe, "Real-time and off-line simulation of a detailed wind farm model connected to a multi-bus network," in *Electrical Power Conference*, pp. 145–152, 2007.
- [80] K. Hopkinson, W. Xiaoru, R. Giovanini, J. Thorp, K. Birman, and D. Coury, "Epochs: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components," in *Power Systems, IEEE Transactions on*, vol. 21, pp. 548–558, 2006.
- [81] L. Hua, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili, "Power system and communication network co-simulation for smart grid applications," in *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, pp. 1–6, 2011.
- [82] J. Nutaro, P. T. Kuruganti, L. Miller, S. Mullen, and M. Shankar, "Integrated hybrid-simulation of electric power and communications systems," in *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, pp. 1–6, 2011.
- [83] A. Sydney, J. Nutaro, C. Scoglio, D. Gruenbacher, and N. Schulz, "Simulative comparison of multiprotocol label switching and openflow network technologies for transmission operations," in *submitted*, 2012.
- [84] S. Das, A. R. Sharafat, G. Parulkar, and N. McKeown, "Mpls-te and mpls vpns with openflow," in *Demonstration at SIGCOMM*, 2011.

- [85] S. Das, A. R. Sharafat, G. Parulkar, and N. McKeown, “Mpls with a simple open control plane,” in *invited talk at Packet Switching Symposium at OFC/NFOEC*, 2011.
- [86] K. B. et al., “Geri-bell labs smart grid research focus: Economic modeling, networking, and security and privacy,” in *SmartGridComm*, 2010.
- [87] N. Cherukuri and K. Nahrstedt, “Cooperative congestion control in power grid communication networks,” in *SmartGridComm 2011*, pp. 587–592, 2011.
- [88] R. B. Bobbo, J. Dagle, E. Heine, H. Khurana, W. H. Sanders, P. Sauer, and T. Yardley, “Enhancing gridwise measurement,” in *Power and Energy Magazine, IEEE*, 2011.
- [89] T. Morris, R. Vaughn, and Y. Dandass, “A retrofit network intrusion detection system for modbus rtu and ascii industrial control systems,” in *System Science (HICSS), 45th Hawaii International Conference on*, pp. 2338–2345, 2012.
- [90] T. Morris, R. Vaughn, and S. D. Yoginder, “A testbed for scada control system cybersecurity research and pedagogy,” in *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research (ACM)*, 2011.
- [91] L. Gang, D. De, and S. W. Zhan, “Smartgridlab: A laboratory-based smart grid testbed,” in *Smart Grid Communications, 2010 First IEEE International Conference on*, pp. 143–148, 2010.
- [92] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, “Nox: Towards an operating system for networks,” in *SIGCOMM CCR*, vol. 38, 2008.
- [93] OpenFlow. OpenFlow Switch Specifications 1.0.0. [Online]: <http://www.openflow.org/documents/openflow-spec-v1.0.0.pdf>, 2010.