
# UNITED STATES MILITARY ACADEMY

**WEST POINT, NEW YORK**

# HONORS THESIS

**Simulating Cyber Conflict Effects on Computer Networks with Python**

by

CDT Stuart R Topp

May 2014

| | |
|---|---|
| Thesis Advisor: | Dr. Chris Arney |
| | Dr. Hilary Fletcher |
| Second Reader: | Dr. Lisa Lowrance |

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | | Form Approved OMB No. 0704-0188 |
|---|---|---|---|



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | May 2014 | Senior Thesis (Honors) |

| 4. TITLE AND SUBTITLE Simulating Cyber Conflict Effects on Computer Networks With Python | 5. FUNDING NUMBERS |
|---|---|
| 6. AUTHOR(S) CDT Stuart R Topp | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br>United States Military Academy<br>West Point, NY 10996-1786 | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)<br>N/A | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER |

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| 12a. DISTRIBUTION / AVAILABILITY STATEMENT | 12b. DISTRIBUTION CODE |
|---|---|
| Approved for public release: distribution is unlimited. | |

**13. ABSTRACT (maximum 200 words)**

A stage oriented, connection-focused approach to examining the effects of cyber conflict on a computer network using Python is presented. The program executes five stages that represent a basic, theoretical cyber conflict exchange. The cornerstone of the program is a connection reliability measurement, which is a number between zero and one that represents the likelihood that two adjacent nodes are able to establish a connection with each other over an edge in the computer network. During each stage, targets are selected based on the ranking of centrality measures, and then their connection reliability measures to adjacent nodes are modified by a multiplier. The multipliers are determined by the relative abilities of the attacker and defender. Edges with reliability ratings that fall below 0.5 are removed from the network at the end of each stage. The program provides a PDF report when the simulation is complete that documents the target selection process, the effects of the reliability measurement multipliers, and the subsequent changes to the network.

| 14. SUBJECT TERMS | 15. NUMBER OF PAGES |
|---|---|
| network analysis, cyber warfare | 71 |
| | 16. PRICE CODE |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UL |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Simulating Cyber Conflict Effects on Computer Networks with Python**

Stuart R Topp
CDT, MI
B.S., United States Military Academy, 2014

Submitted in partial fulfillment of the
requirements for the degree of
BACHELOR OF SCIENCE
in **Operations Research**
with Honors
from the
UNITED STATES MILITARY ACADEMY
May 2014

Author: Stuart R Topp

Approved by:

Doctor Chris Arney
Thesis Advisor

Doctor Hilary Fletcher
Co-Advisor

Doctor Lisa Lowrance
Second Reader

Colonel Michael Phillips
Chairman, Department of Mathematical Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

A stage oriented, connection-focused approach to examining the effects of cyber conflict on a computer network using a Python simulation is presented. The program executes five stages that represent a basic, theoretical cyber conflict exchange. The cornerstone of the program is a connection reliability measurement, which is a number between zero and one that represents the likelihood that two adjacent nodes are able to establish a connection with each other over an edge in the computer network. During each stage, targets are selected based on the ranking of centrality measures, and then their connection reliability measures to adjacent nodes are modified by the attack's effectiveness. The multipliers are determined by the relative abilities of the attacker and defender. Edges with reliability ratings that fall below 0.5 are removed from the network at the end of each stage. The program provides a report when the simulation is complete that documents the target selection process, the effects of the reliability measurement multipliers, and the subsequent changes to the network.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF FIGURES

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# ACKNOWLEDGMENTS

I would like to thank my advisory team, Dr. Arney and Dr. Fletcher. Dr. Arney was extremely patient while I recovered from a concussion during the first few months of my research, and then helped me focus my work once I recovered so I could complete my research in time.

THIS PAGE INTENTIONALLY LEFT BLANK

# EXECUTIVE SUMMARY

This research developed a Python-based simulation program that utilized a five-stage/two-actor method for modeling cyber conflict. The five stages reduce cyber conflict to: battlespace preparation, first attack wave, defender's response, second attack wave, counterattack, and recovery. Within the simulator, the attacker and defender can have a high, medium, or low skill level for each stage. This research also developed a connectivity measure for each edge in a computer network that represents the probability that a packet sent from a source actually arrives at its recipient. The simulator models cyber conflict effects by selecting and modifying connectivity measures based on user-defined settings. The targeting algorithms use centrality lists to select prominent nodes within the network. Effects are modeled by manipulating edge connectivity measures, depending on the skill level combination for each stage. The simulator was tested using nine data points, where the attacker and defender each used the same skill level across all five stages, using a basic sample network from a 2009 cadet cyber exercise. The results indicated that the simulator accurately followed the five-stage model. When attackers overmatched the defenders, the network's connectivity became significantly degraded, while when the defenders overmatched the attackers, the network connectivity was only slightly degraded. Future work should validate the simulated network degradation using actual cyber conflict data, and address several design flaws identified during the test.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

Cyber is the newest, and perhaps the most pervasive, battlespace. This battlespace encompasses more than just military information systems and networks. National infrastructure, the finance industry, and all of the world's "connected" civilian populations are, in one way or another, a part of the military's Cyber domain. Therefore, Cyber has the potential to affect billions of lives on a daily basis. Society's growing dependence on cyber support makes it a very attractive target for criminals, terrorists, and competing nations. Cyber activities can range from "hacktivism" and electronic vandalism, to crime directed at private citizens, private industry, or government agencies, to covert "grey" state-sponsored espionage or attacks, to overt cyber attacks in conjunction with military operations or infrastructure utility. This chapter introduces the elements of cyber science, and provides a brief overview of network science.

## A. WHAT IS CYBER?

Richard Clark, a former National Coordinator for Security, Infrastructure Protection, and Counter-terrorism for the United States under Presidents George H.W. Bush, Bill Clinton, and George W. Bush, and Robert Knake define cyberspace as "all of the computer networks in the world and everything they connect and control," including private networks, supervisory control and data acquisition systems, and any "information managed by [a] computer network" (2010, p. 70). Daniel T. Kuehl, a professor at the National Defense University and a former nuclear planner for the United States Air Force, in his contribution to *Cyberpower and National Security* (Kramer, F.D., Starr, S.H., & Wentz, L., 2009) similarly defines cyberspace as:

> a global domain within the information environment whose distinctive and unique character is framed by the use of electronics and the electromagnetic spectrum to create, store, modify, exchange, and exploit information via interdependent and interconnected networks using information-communication technologies. (as cited in Gray, 2013, p. 9)

Andrew F. Krepinevich, a defense policy analyst, provides a more abstract and Simple English definition of cyberspace in *Cyber Warfare*. According to Krepinevich:

> Cyberspace comprises all of the world's computer networks, both open and closed, to include the computers themselves, the transactional networks that send

data regarding financial transactions, and those networks comprising control systems that enable machines to interact with one another. (as cited in Gray, 2013, p. 9)

Cyber science is new, and the civilian and military perceptions and usage of information technology are steadily evolving each year, as a steady stream of new technological advances continues to change the information technology world and its integration into daily life and military equipment and operations. Cyber science is too new, and evolving too rapidly to have a clear, academically agreed upon definition. For the purposes of this research, cyberspace will refer to the world's physical and virtual networks—both public and private; the hardware, communication systems, and information systems connected to those networks; and the data stored on those systems. This definition is meant to be as inclusive and all-encompassing as possible in order to limit the potential for relevant systems and information to be excluded.

## B. NETWORK SCIENCE OVERVIEW

At the most abstract level, network science "is the organized knowledge of networks based on their study using the scientific method" (Committee on Network Science for Future Army Applications, 2005, p. 26). Network science draws on theories from many disciplines, including graph theory and social science, and can be applied to almost any discipline where there are actors or other "things" and relationships, or links, between them (Lewis, 2009).

Graph theory provides the basic foundation for modeling a network:

$$G = \{N, L, f\}$$

"where $N$ is a set of nodes, $L$ a set of links, and $f: N \times N$ a mapping function that defines the structure of G—how nodes are connected to each other through links" (Lewis, 2009, p. 6). When the model incorporates time, $G$ becomes:

$$G(t) = \{N(t), L(t), f(t): J(t)\}$$

where $t$ is time, $N$ is a set of nodes (also referred to as vertices or actors), $L$ is a set of links (also referred to as edges), $f: N \times N$ is a mapping function that connects nodes, and $J$ is an "algorithm for describing [the] behavior of nodes and links versus time" (Lewis, 2009).

Lewis identifies eight general principles of network science: structure, emergence, dynamism, autonomy, bottom-up evolution, topology, power, and stability. First, "networks have structure," and "are not random collections of nodes and links" (p. 19). Second, "a network property is emergent if it changes by a factor of 10 as a consequence of a dynamic network achieving stability" (p. 19). Third, a network's "dynamic behavior is often the result of emergence or a series of small evolutionary steps leading to a fixed-point final state of the system" (p. 20). Fourth, "a network forms by the autonomous and spontaneous action of independent nodes that 'volunteer' to come together (link), rather than through central control or central planning" (p. 20). Fifth, "networks grow from the bottom or local level up to the top or global level" (p. 20). Sixth, "the architecture or topology of a network is a property that emerges over time as a consequence of distributed—and often subtle—forces or autonomous behaviors of its nodes" (p. 21). Seventh, "the power of a node is proportional to the number and strength of its nodes and links" (p. 21). Finally, "a dynamic network is stable if the rate of change in the state of its nodes/links or its topology either diminishes as time passes or is bounded by dampened oscillations within finite limits" (p. 21).

Depending on the discipline utilizing network science, $N$ can be a set of actors, vertices, points, nodes, or agents, and $L$ can be a set of the relationship between actors, edges, links, or connections. There are several ways to measure these connections. For the purposes of the following definitions, $N$ will be a set of nodes, and $L$ will be a set of edges. The **degree** of a node is the sum of the links connecting that node to the graph, and whichever node has the largest degree in a graph is called the **hub**. The longest path from a node to all the other nodes is the **radius** of the node, and the **diameter** of a graph is the length of the longest radius (whose respective nodes are called **peripheral nodes**), while the **center** of the graph is the node or nodes with the shortest radii. The **betweenness** of a node is the number of paths that link all of the other nodes to each other that pass through the original node. The **closeness** of a node is the number of direct paths that link all of the other nodes to each other that must pass through the original node (Lewis, 2009). These measures are used to identify relationships and the relative level of importance of different nodes within the graph.

Cyber science encompasses the underlying theory and structure of the cyber domain: information systems, the data on those systems, communications and processes between systems, etc. Cyber scientists can use network and systems science to mathematically describe the disposition of information systems and their networks, identify relationships between information systems, identify critical nodes, and determine the broader structural effects of modifications to elements of information systems.

# II. CYBERSPACE AND CYBERWAR

The first network was created in 1965, when Lawrence Roberts and Thomas Merril connected two computers in Massachusetts and California to each over a telephone connection. This led to the creation of the Advanced Research Projects Agency Network (ARPANET) in 1969, which connected computers at various universities throughout the United States. Several more networks emerged, and by 1990, ARPANET was decommissioned, having been replaced by dozens of independent networks, and the World Wide Web emerged in 1992 (Leiner et al., n.d.). In less than fifty years, computer networks have evolved from the exclusive domain of elite research universities and the military, to an integral part of everyday life for much of the world.

## A. A BRIEF HISTORY OF CYBER CONFLICT

Computer malware has been an ever-present nuisance in cyberspace since the early days of ARPANET. The threat to information systems has only increased as technology has become more ubiquitous and integrated into daily life. Jason Healey, in conjunction with the Atlantic Council and the Cyber Conflict Studies Association, recently published the first ever cyber conflict history book in 2013. The next two sections will provide a brief history of cyber history, identify evolving threats, and explain Healey's "three lessons" that policymakers have not yet learned from repeated attacks and cyber events.

### 1. History

Healey identified seven "Cyber Wake-Up Calls:" the Morris Worm, Operations ELIGIBLE RECEIVER, SOLAR SUNRISE, MOONLIGHT MAZE, and BUCKSHOT YANKEE, Chinese Espionage, Estonia and Georgia, and Stuxnet. These incidents range from a relatively harmless and easy to defeat computer virus (the Morris Worm) to National Security Agency (NSA) red teams testing Department of Defense (DoD) networks (ELIGIBLE RECEIVER), to all-out cyber-attacks by "patriot hackers" who may or may not have been operating with the support of a nation-state while that nation-state conducted kinetic operations (Georgia) (Healey, 2013).

The Morris Worm was released in 1988, and "was deliberately designed to do two things: infect as many machines as possible, and be difficult to track and stop" (Spafford, 1988, p. 26). The worm crashed a tenth of the existing Internet, but resulted in a

successful private sector response, and demonstrated that the government does not have the expertise or dexterity to respond to major cyber incidents. Although the Morris Worm resulted in the creation of Carnegie Mellon University's Computer Emergency Response Team (CERT), which was funded by the DoD (Healey, 2013), the private sector would remain firmly in control of the United States' response to cyber incidents that do not exclusively affect military or governmental systems, as Mark Bowden highlighted in his 2011 book on the Conficker Worm. During the civilian-dominated response to Conficker, civilian cyber security experts were only able to brief the worm's development to government officials several months after the worm had emerged as a new and potentially significant threat. Even after the government became involved, the official response was insignificant in comparison to that of the private sector.

Operation ELIGIBLE RECEIVER was a "no notice interoperability exercise" conducted by the NSA in 1997, where thirty-five hackers repeatedly infiltrated DoD networks, and were detected only twice, resulting in the acceleration of the implementation of cyber reform policy within the DoD, but did not result in any new policy development (Hildreth, as in Healey, 2013, Part 1, Takeoff, para. 6). Operation MOONLIGHT MAZE, which began in March 1998, was a non-attributed intrusion into non-classified networks, which resulted in the theft of sensitive defense research information. The Russians were suspected to be behind the intrusions, but this was never publically proven. Operation TITAN RAIN, a set of intrusions that targeted several US government agencies, American contractors, and the Dalai Lama, among others, was publically attributed to China in 2005. This operation set involved the theft of information on the F-35 Joint Strike Fighter. Finally, in Operation BUCKSHOT YANKEE, intruders gained access to the Secret Internet Protocol Router Network (SIPRNET) and the Joint Worldwide Intelligence Communications System (JWICS) by "jumping the air gap" between these classified networks through flash drives. Analysts suspected that Russia was again behind this intrusion, but this was never publically confirmed (Healey, 2013).

The best-documented and most recent "total cyber conflict" is the 2008 Russian-Georgian conflict, where DDoS attacks, intrusions, and defacements forced important government and news sites offline in conjunction with a Russian military invasion

(Healey, 2013). “Georgia effectively lost control of the nation’s ‘.ge’ domain,” while “the Georgian banking sector shut down,” triggering a collapse in credit card payment and mobile phone systems (Clark & Knake, 2010, pp. 19-20). The attacks were clearly attributed to Russian patriotic hackers, who were assisted by Russian organized criminals. Analysts were unable to prove whether the Russian government was involved in any capacity beyond condoning the attacks (Healey, 2013).

The most recent “Cyber Wake-Up Call” was the Stuxnet virus, which was discovered in 2010. Stuxnet is a cyberweapon allegedly developed in a partnership between Israel and the United States, and targeted the specific supervisory control and data acquisition (SCADA) industrial control systems (ICS) used by Iranian centrifuges (Healey, 2013). The virus jumped the air gap between the Internet and the Iranian nuclear ICS, presumably through flash drives. Once Stuxnet reached the SCADA device it was designed for, it manipulated the rotation frequencies of the targeted centrifuges, damaging them beyond repair. The United States and Israel maintain that Stuxnet set back the Iranian nuclear program by several months or years, but independent analysts note that Iran’s uranium-enrichment capacity actually increased during the Stuxnet attacks (Barzashka, 2013).

This brief history of cyber conflict is by far not all-inclusive, but demonstrates the evolution of cyber conflict and cyber tools over the last fifty years. Cyberspace has been used for everything from simple pranks gone wrong, to industrial and military espionage, to widespread denial of service attacks in conjunction with military operations, to target-specific viruses designed to destroy nuclear equipment without a trace. Because of the broad nature of cyberspace, both military and civilian critical infrastructures are vulnerable to the same style of attacks, which are much easier to conduct than they are to prevent or attribute.

### 2. Takeaways

Healey identifies three key takeaways from his documentation of cyber conflict’s history. These are:

1. Cyber conflict has changed only gradually over time; thus historical lessons derived from past cases are still relevant today (though these are usually ignored).

2. The probability and consequences of disruptive conflicts have often been hyped, while the real impacts of cyber intrusions have been consistently under-appreciated.
3. The more strategically significant a cyber conflict is, the more similar it is to conflicts on the land, in the air, or on the sea. (Healey, 2013, Introduction, paras. 5-7).

These takeaways indicate that policymakers need to develop updated, relevant cyber policies and strategy that reflect an anticipation of, instead of a reaction to, events similar to those of the past. Threats in cyberspace are multifaceted, and do not come from just nation-states, or terrorists, or any other singular threat. Teenagers, patriot hackers, criminals, foreign companies, intelligence services, and militaries are all capable of similar attacks but on varying scales. A close examination of known, past threats, can help shape and anticipate future threats, but only if policies are developed proactively.

## B. CYBER "WAR"

Richard Clark and Robert Knake's *Cyber War* identifies five vulnerabilities in the basic design of the internet, and proposes a "Defensive Triad" strategy create federal regulations that forces the American section of cyberspace to adopt modern and effective security policies.

### 1. Vulnerabilities of the Internet

The five vulnerabilities that Clark and Knake identify are: the Internet's addressing system, the lack of Internet governance, the use of unencrypted transmissions, unrestricted malware traffic, and the decentralized design of the Internet. Together, these vulnerabilities exacerbate the Internet's existence as a "neutral medium," and makes it easier for both legitimate users and malicious hackers to take advantage of the Internet (2010, p. 83).

The Internet's addressing system has two components: the Domain Name Server (DNS), which translates web addresses into numbers, and the Border Gateway Protocol (BGP), which functions as the post office of the Internet. DNS vulnerabilities provide cyber warriors with the opportunity to misdirect traffic and potentially shut down the World Wide Web through a Distributed Denial of Service (DDoS) attack, which actually

occurred in February 2007. The BGP can potentially be spoofed, resulting in Internet traffic "get[ting] lost and not reach[ing] its destination" (Clark & Knake, 2010, p. 78).

The second identified vulnerability is the lack of Internet governance. There is no "network administrator" that ensures all of the parts of the Internet function as they are intended to, and that they follow all rules and guidelines (Clark & Knake, 2010, p. 79). The third vulnerability is the use of unencrypted transmissions throughout the Internet. Although most web sites use an encrypted connection to transmit login and payment information, the majority of web browsing is done "in the clear," making those transmissions vulnerable to the Internet's version of wiretapping, called "packet sniffing" (Clark & Knake, 2010).

Malware is a collective term for the malicious code that is transmitted across the Internet and executed on computers and other information systems and network components. The Internet's fourth vulnerability is the blind transmittal of traffic. The Internet's communication protocols only look at packet headers, or the "to" and "from" lines, and Internet Service Providers (ISPs) generally do not monitor the content of the traffic that passes through their networks. This places the burden of protection onto the end user, which is ineffective when the majority of end users does not understand or care about cyber threats (Clark & Knake, 2010).

The final vulnerability is the decentralized design of the Internet, which is directly related to the lack of Internet governance mentioned above. There is no control over the direction or usage of the Internet. No single force can institute a change across the entire Internet (Clark & Knake, 2010). Any policy decisions to make the Internet more secure must be made by a national government, and those decisions only affect that country's small corner of the Internet, assuming that residents comply with new policies.

### 2. "The Defensive Triad"

"The Defensive Triad Strategy would use federal regulation as a major tool to create cyber security requirements, and it would, at least initially, focus defensive efforts on only three sectors" (Clark & Knake, 2010, p. 160). These three sectors are the "backbone of the Internet", the power grid, and the DoD. According to Clark, regulating and improving these three sectors will significantly improve the United States

government's ability to defend itself and America's critical infrastructure from cyber threats.

The Internet backbone consists of the Tier 1 ISPs that moves 90% of U.S. Internet traffic. Clark proposes that inspecting Internet traffic before it connects to the backbone would allow cyber defenders to stop attacks before they could reach their target. The technology for this kind scanning currently exists, and the scans can be completed without decreasing network speeds, but it would need to be developed further in order to keep pace with advances in Internet speed. This monitoring would need to be done by an independent party, and would face many policy hurdles and need to overcome widespread privacy concerns (Clark & Knake, 2010), especially after the recent NSA leaks that revealed widespread government snooping and data collection, which resulted in a massive lost of public trust and calls for restraints on government surveillance programs (BBC, 2013).

The American power grid is currently accessible, indirectly, via the Internet, and some control systems broadcast commands using unencrypted radio transmissions. Clark proposes that the Department of Energy (DoE) develop regulations that require power grid control systems to be inaccessible from the Internet, for commands to be encrypted before being broadcast, and for systems to authenticate commands before accepting them (2010).

Finally, Clark recommends five additions to the DoD's current security upgrade plan:

- Install desktop firewalls and antivirus and intrusion-prevention software on all computers on all DoD networks, whether or not they are connected to the Internet;
- Require all users on all DoD networks to prove who they are when they sign on through at least two factors of authentication;
- Segment the networks into subnets with limited "need to know" access rules for connecting out of the subnets;
- …encrypt all files on all computers…
- Monitor all networks for new unauthorized connections to the network, automatically shutting off unknown devices. (2010, p. 174)

In other words, this plan calls for a "defense-in-depth" of the DoD network. Based on accounts of previous attacks and infiltrations of the DoD network, the current "perimeter defense" techniques used by the DoD are ineffective at identifying threats and preventing them from entering and exploiting air gapped networks.

## C. PENETRATION TESTING

Penetration testing, also known as ethical or white hat hacking, is "a legal and authorized attempt to locate and successfully exploit computer systems for the purpose of making those systems more secure" (Engebretson, 2011, p. 1). The only difference between penetration testing and malicious hacking is the intent behind the penetration and the effects of the exploitation of a system. Both ethical and malicious hackers use the same tactics, techniques, and procedures to penetrate and exploit systems.

Engebretson teaches a four step process in his 2011 book, *The Basics of Hacking and Penetration Testing*. His technique, which he refers to as the Zero Entry Hacking Penetration Testing Methodology, has four stages: reconnaissance, scanning, exploitation, and maintaining access. These four stages are cyclical: after a system has been exploited, it may present additional targets.

The first stage, reconnaissance, can be broken down into two strategies: active and passive reconnaissance. An active strategy requires "interacting directly with the target," who may "log our activity," while a passive strategy simply utilizes information in the public domain, and does not directly interact with the target (Engebretson, 2011, p. 18). A thorough reconnaissance of the target should yield specific systems and access points for scanning.

The second stage, scanning, has three steps: "determining if a system is alive," "port scanning the system," and "scanning the system for vulnerabilities" (Engebretson, 2011, p. 44). These steps identify active systems, find open communication points, or ports, and then determines what exploits are appropriate for the specific system. The third stage, exploitation, "is the process of gaining control over a system," with the desired result being "administrative-level access to the computer" (Engebretson, 2011, p. 65). There are countless ways to take over a system. As software is patched and hardware upgraded, some exploits become extinct, but these upgrades themselves often produce their own, new exploits. The final step is maintaining access. This is done through the

establishment of a backdoor program and by installing rootkits, or programs that operate within the computer kernel and are very difficult to detect (Engebretson, 2011).

# III. METHODOLOGY

To date, there has been no known active cyber conflict, where combatants on both sides actively sought to degrade or destroy each other's information systems capabilities, and there is no accompanying body of sound, strategic though on cyber conflict. Policymakers need tools to develop sound and legitimate cyber policies based on possible actions and outcomes, and not derived from policies in other battlespaces. During any conflict, maintaining communication and the flow of information is critical for coordination and success. In a cyber conflict, communication and connectivity between information systems and users are especially important, because without either, systems lose their value.

## A. POLICY EXPLORATION

I developed a five stage model for cyber conflict based on an amalgamation of Engebertson's methodology and historical case studies, outlined in Table 1, below. The first stage is the battlespace preparation stage, where both sides have the opportunity to prepare the battlespace by emplacing logic bombs, backdoors, and other malicious code within their opponent's networks. Based on each side's internal security posture, there is a possibility that some or all of these preparations might be discovered. The second stage is the first wave of attacks, where the attacker initiates his initial exploits, and the defender's network responds automatically, if at all. The third stage is the first response stage, where the defender develops a deliberate response to the specific exploits launched against his network, but the attacker is able to respond in real-time. The fourth stage consists of two parts. In the first part of Stage IV, the attacker launches his second wave of exploits, while the defender responds in real-time. In Stage IVa, the defender launches a counterattack (as dictated by the defender's cyber policy). In the final recovery stage, both sides rebuild or enhance their capabilities. This methodology is a simplification of the dynamic and real-time realm of cyber conflict, but, due to the limited pool of data to draw from, must currently suffice.

| Stage | Stage I | Stage II | Stage III | Stage IV | Stage IVa | Stage V |
|---|---|---|---|---|---|---|
| Name | Battlespace Preparation | First Wave | First Response | Second Wave | Counterattack | Recovery |
| Events | Preparation of the battlespace | Initial exploits initiated | Defender's deliberate responses | Second wave of exploits initiated | Defender initiates exploits | Both sides rebuild or enhance capabilities |
| | Emplacement and discovery of malicious code | Automated network responses | Automated attacker network responses | Automated network responses | Policy dependent | |
| | Active/passive security | | Real-time attacker responses | Real-time defender responses | Automated and real-time attacker responses | |

**Table 1: Simulation Stages**

## B. SIMULATION DESIGN AND ALGORITHM DEVELOPMENT

I developed a simulation program in Python called the Cyber Conflict Network Simulator (CCNS), based partially on the Python-based Advanced Network Targeting (ANAT) social network analysis program (Johnson, McCulloh, Curwin, & Topp, 2013). The ANAT program reads in a network from a comma separated value file, builds the component networks and metanetwork, uses bridging to create social networks, and then produces a targeting report. The CCNS borrows the ANAT's module for reading the source file and building the metanetwork and uses the same modules for calculating network centrality and centralization measures, but the similarities end there. After creating the metanetwork, the CCNS executes a module for each simulation stage, based on the designated skill level of the attacker and defender. The skill levels are found in Table 2, below. Currently, there has not been any development of the Stage IVa module, and Stage V only has one "standard" level.

In order to capture the cumulative effects of cyber conflict on a network's connectivity, I developed a connectivity measure, or "connectivity probability," that represents the likelihood that a message sent from an originator arrives at the recipient. The connectivity probability exists for each edge in a given network, and can be different

for messages traveling in opposite directions. This measure combines the results of various cyber attack techniques, and generalizes conflict in order to allow the simulator to work for most networks, instead of becoming highly tailored for specific types of networks and conflicts.

| Factor | | | Levels | | |
|---|---|---|---|---|---|
| | | | Low | Medium | High |
| Stage I | Battlespace Preparation | Attacker | None | Limited | Active |
| | | Defender | Passive | Limited | Active |
| Stage II | First Wave | Attacker | Basic | x | Advanced |
| | | Defender | None | Basic | Advanced |
| Stage III | First Response | Attacker | None | Basic | Advanced |
| | | Defender | Basic | x | Advanced |
| Stage IV | Second Wave | Attacker | Basic | x | Advanced |
| | | Defender | None | Basic | Advanced |
| Stage IVa | Counterattack | Attacker | None | Basic | Advanced |
| | | Defender | Basic | x | Advanced |
| Stage V | Recovery | Attacker | None | Basic | Advanced |
| | | Defender | None | Basic | Advanced |

**Table 2: Simulation Factors and Levels**

### 1. Simulation Algorithms

Each simulation stage follows the same pattern. First, the stage selects a target or target list. Then, the simulator identifies the location of each target in the incidence and connectivity value matrices. Next, the simulator "attacks" each target by locating the edges connected each target, and multiplies the connectivity value for each edge by a given multiplier that represents the net effects of the stage's cyber attacks and the defenders' repairs. If any connectivity values are reduced to below 0.5, the edge is considered to be no longer reliable, and the edge is removed from the network. If the connectivity value of an edge that was previously removed from the network is restored above the 0.5 threshold, the edge is added back to the network. Finally, the simulator calculates the new network centralities.

Target selection is based on node centrality measurements (calculated at the end of the previous stage). One or more targets may be selected. When a single target is selected, the node with the highest selected centrality becomes the target. Degree centrality selects the most connected target, while betweenness centrality identifies gate-

keepers within the network, closeness centrality indicates how "close" a node is to the other nodes in the network, and Eigenvector centrality identifies nodes connected to other, important nodes (McCulloh & Johnson, 2011). When multiple targets are required, a given number of nodes with the highest indicated centralities are selected (a node can only be selected as a target once, even if it appears on the top of multiple centrality lists). In Stage V, targets can be selected from the list of edges removed from the network, or from a list of edges with the lowest connectivity values (and therefore need to be restored), and the multipliers are always greater than one.

Multipliers are used to modify the network connectivity values. The multipliers for each stage are determined based on the quotient of the defender's and attacker's skill level values. The skill level values are numerical approximations for the skill level of the attacker or defender. The formula for determining the multiplier is:

$$M = c * \frac{D}{A}$$

where $M$ represents the multiplier, $D$ represents the defender's skill level, $A$ represents the attacker's skill level, and $c$ is a constant. For example, a defender with a skill level of 3, an attacker with a skill level of 5, and a constant of 1 yields a stage multiplier of 0.6. Each stage has a unique constant that reflects the nature of the cyber activities during that stage.

## C. TESTING THE CYBER CONFLICT NETWORK SIMULATOR

Since I was unable to obtain actual data detailing the effects of a cyber conflict, I validated the general effects of the CCNS using a sample computer network based on the 2009 Cadet Cyber Exercise. The network (Figure 1, page 17) represents the network used on the first day of the exercise, and is representative of a small network that would be used by military cyber defenders ("CDX 2009 network USMA," 2009). The network has 26 nodes and 47 edges. The degree centralization is 0.2733, the betweenness centralization is 0.543, and the closeness centralization is 0.2881. The network's density is 0.0723, and the diameter of the network is 9.

There are 1,994 possible combinations of levels across the four currently developed stages (Stage IVa has not been developed yet, and Stage V currently only has one setting). To reduce this to a more manageable level, I selected nine data points

spanning the set of all possible combinations of simulations where the skill level of the attacker and defender remained constant across each stage (Table 4, page 18). When a medium setting was not available for a stage, the high setting was used. This is a fair method of assessing the simulator's ability to simulate conflict where the attacker and defender maintain their skill level throughout the conflict, but does not validate situations where the attacker or defender might be better skilled at one stage over another.

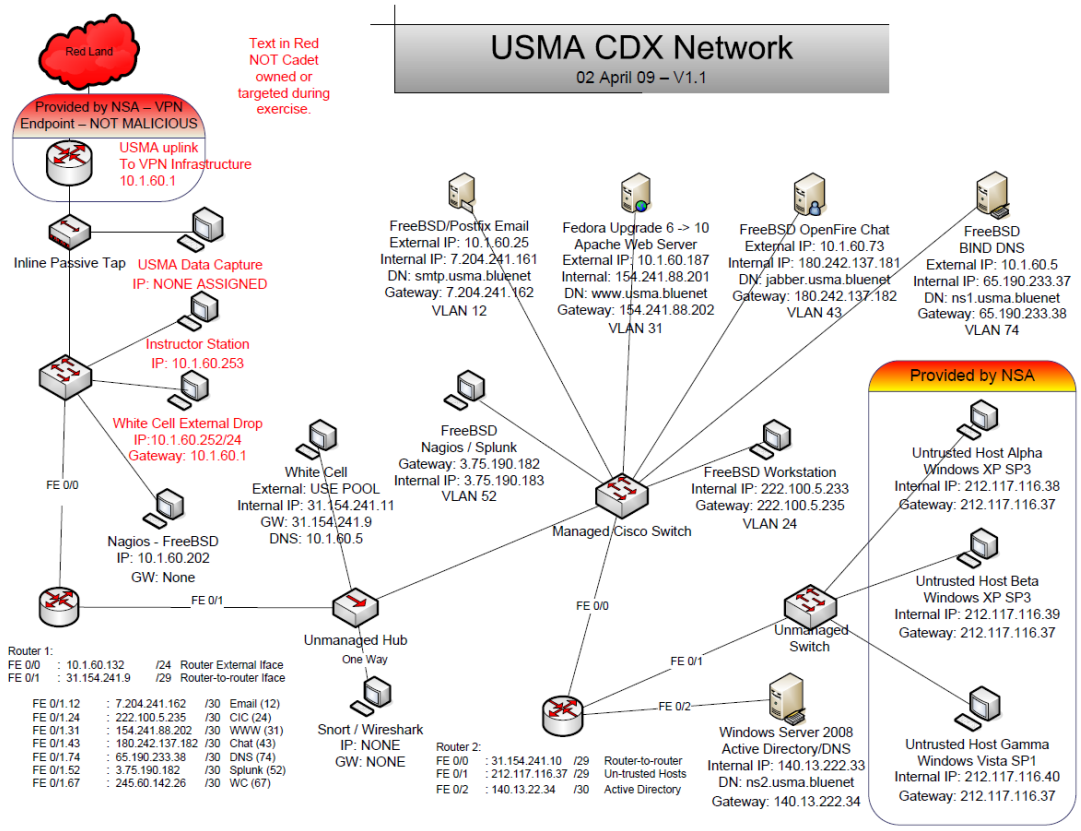


**Figure 1: Initial Network Diagram, 2009 Cadet Cyber Exercise.**

I assigned the following values to the low, medium, and high settings in order to calculate the multipliers:

| Skill Level | Skill Value |
|---|---|
| Low | 1 |
| Medium | 5 |
| High | 10 |

**Table 3: Skill Levels and Corresponding Values.**

I then estimated constants for each stage and skill level combination in order to achieve a multiplier that I estimated would produce the appropriate effects on the network. The multiplier calculation tables are located in Appendix A, which begins on page 34.

| | Stage 1 | | Stage 2 | | Stage 3 | | Stage 4 | | Stage 5 |
|---|---|---|---|---|---|---|---|---|---|
| **Data Point** | **Attacker** | **Defender** | **Attacker** | **Defender** | **Attacker** | **Defender** | **Attacker** | **Defender** | **All** |
| **1** | Low | Low | Low | Low | Low | Low | Low | Low | Standard |
| **2** | Low | Medium | Low | Medium | Low | High | Low | Medium | Standard |
| **3** | Low | High | Low | High | Low | High | Low | High | Standard |
| **4** | Medium | Low | High | Low | Medium | Low | High | Low | Standard |
| **5** | Medium | Medium | High | Medium | Medium | High | High | Medium | Standard |
| **6** | Medium | High | High | High | Medium | High | High | High | Standard |
| **7** | High | Low | High | Low | High | Low | High | Low | Standard |
| **8** | High | Medium | High | Medium | High | High | High | Medium | Standard |
| **9** | High | High | High | High | High | High | High | High | Standard |

**Table 4: Design of Experiment for Testing the USMA CDX Network.**

# IV. RESULTS

I executed one run of each data point in the CCNS, and saved the results to a comma separated value file in order to compile and analyze the outcome of each stage. Since there are no stochastic elements to my model, each run of the same data point produces the same data. I exported the network's centrality measures and connection strengths after each stage, and a list of the deleted links, beginning with "Stage 0" or the initial network. I then calculated the average centrality measures for each stage:

$$ACM_k = \frac{\sum_{i=1}^{n} C_i}{n}$$

where $ACM_k$ is the average centrality measure (degree, closeness, betweenness, and eigenvector) of Stage $k$, $C_i$ is the centrality of the $i^{th}$ node in the network, and $n$ is the total number of nodes in the network. I also calculated the change in average centrality measures between each stage:

$$ACM_{\Delta k,k-1} = ACM_k - ACM_{k-1}$$

where $ACM_{\Delta k,k-1}$ is the change in average centrality measure between Stages $k$ and $k$-1.

I then calculated the change in connectivity measure for each node between stages:

$$Con_{\Delta i,k,k-1} = Con_{i,k} - Con_{i,k-1}$$

where $Con_{\Delta i,k,k-1}$ is the change in connectivity for node $i$ between Stages $k$ and $k$-1, and $Con_{i,k}$ is the connectivity value for node $I$ after Stage $k$. I noticed that a bug in the simulator caused some connectivity values to become greater than one. I rounded those values down to 1 in order to maintain the connectivity measure's probabilistic definition.

I calculated the aggregate average centrality measures for each Stage $k$:

$$AACM_k = \frac{\sum ACM_{m,k}}{m}$$

where $AACM_k$ is the average of the average centrality measures, and $m$ is the number of design points. I then calculated the changes in average centrality measures between each stage:

$$AACM_{\Delta k,k-1} = AAMA_k - AAMA_{k-1}$$

where $AACM_{\Delta k,k-1}$ is the change in the selected aggregate average centrality measure.

I calculated the average connectivity values for each node *i* after Stage *k* across all of the design points:

$$ACon_{i,k} = \frac{\sum Con_{i,k}}{m}$$

where $ACon_{i,k}$ is the average connectivity value for node *I* after Stage *k*. Finally, I calculated the changes in average connectivity values for each node between each stage:

$$ACon_{\Delta i,k,k-1} = ACon_{i,k} - ACon_{i,k-1}$$

where $ACon_{\Delta i,k,k-1}$ is the change in average connectivity value for node *i* between stages *k* and *k*-1.

**Design Point 1** (attacker: low; defender: low) resulted in minor network connectivity degradation, but never to the point where an edge became ineffective and

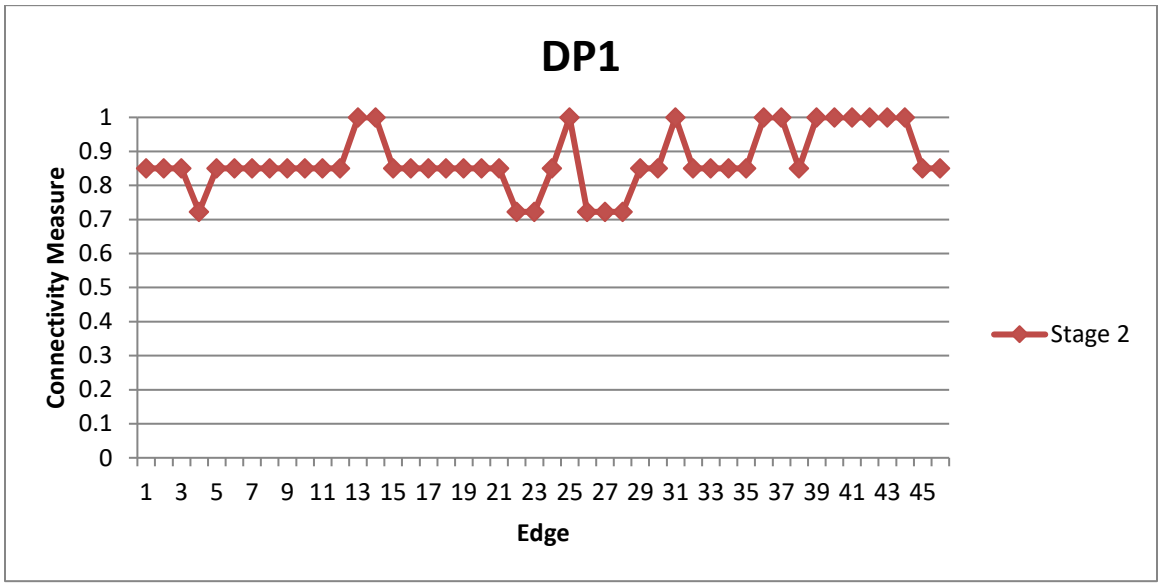


**Figure 2: Design Point 1, Stage II: Connectivity Measures.**

was removed from the network. Figure 2, above, shows the connectivity measures for each edge during Stage II, when the connectivity measures were the most degraded. No connectivity values fell below 0.7 during the simulation. Figure 3 shows the average centrality measures across all five stages during Design Point 1. The average centrality measures never changed, indicating that no edges were removed from the network, and the structure of the network did not change.

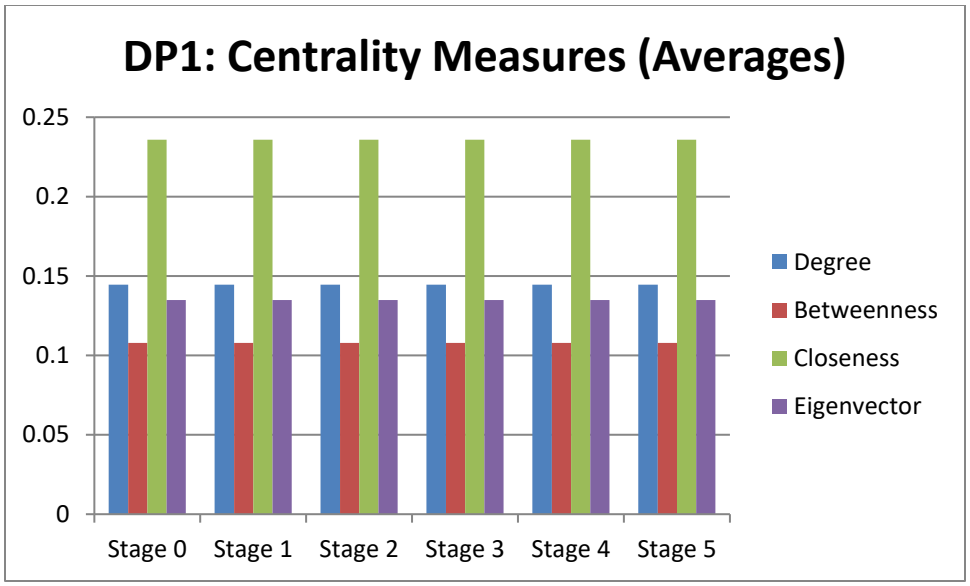


**Figure 3: Design Point 1: Average Centrality Measures.**

**Design Point 2** (attacker: low; defender: medium) resulted in very minor network connectivity degradation. In this case, no connectivity values fell below 0.8 during the simulation, and therefore, no edges were removed from the network. Figure 4, below, shows the connectivity measures for each edge during Stage II, when the connectivity measures were the most degraded. No connectivity values fell below 0.8 during the simulation. The graph of average centrality measures for Design Point 2 is the same as that of Design Point 1, above.

**Design Point 3** (attacker: low; defender: high) resulted in negligible network connectivity degradation. In this case, no connectivity values fell below 0.9 during the simulation, and no edges were removed from the network. Figure 5, below, shows the connectivity measures for each edge during Stage II, when the connectivity measures were most degraded. The graph of average centrality measures for Design Point 3 is the same as that of Design Point 1, above.

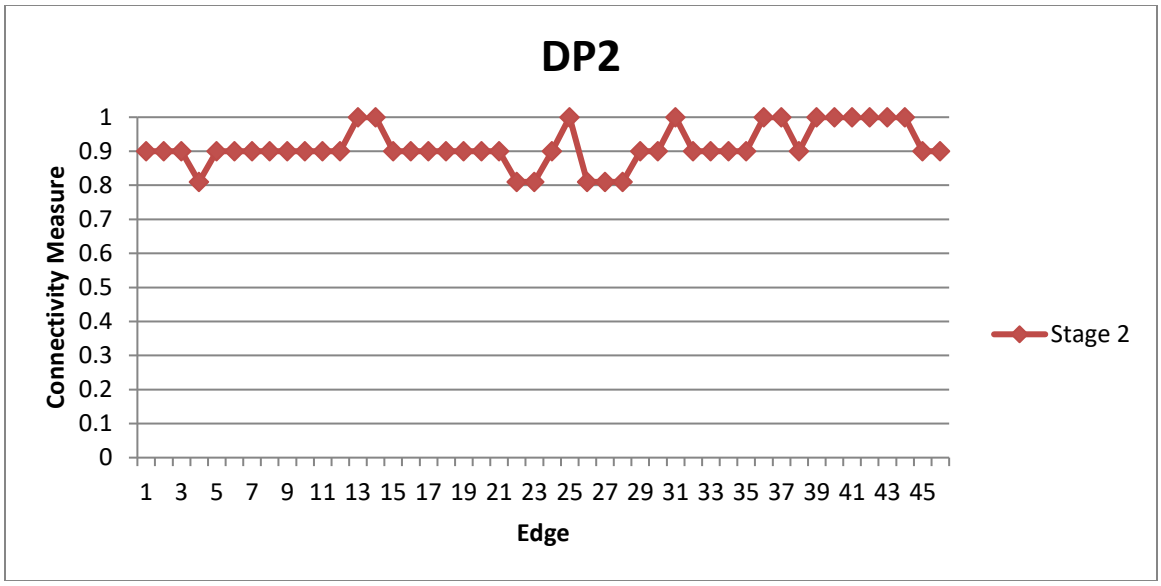


**Figure 4: Design Point 2, Stage II: Connectivity Measures.**

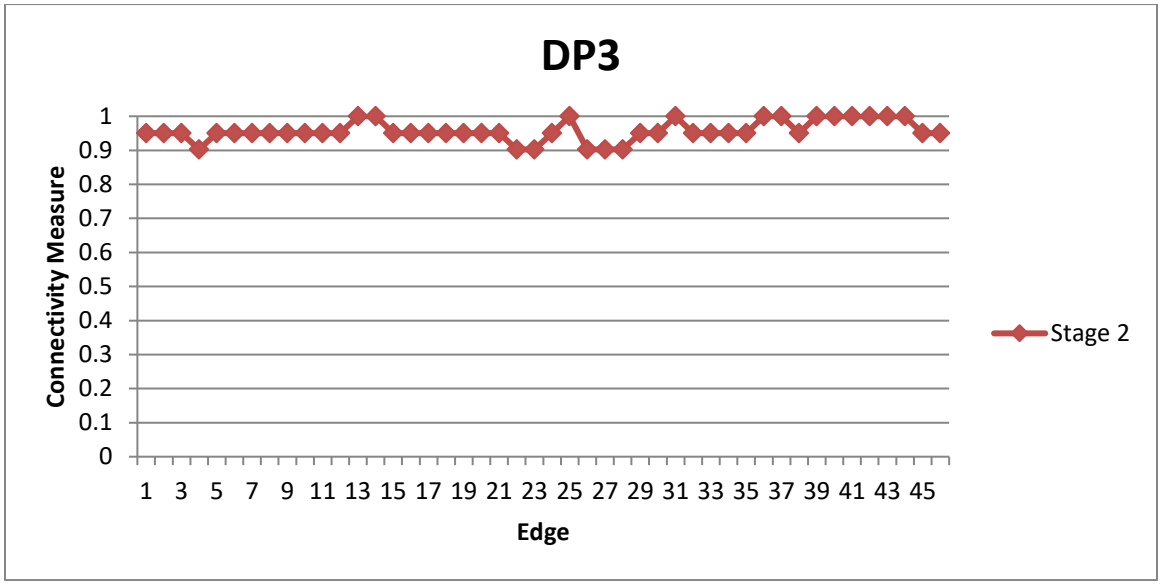


**Figure 5: Design Point 3, Stage II: Connectivity Measures.**

**Design Point 4** (attacker: medium; defender: low) resulted in significant network connectivity degradation. I have included all of the connectivity measure graphs for this design point in Appendix B, which begins on page 36. In Stage I, approximately one third of the edges' connectivity values fell to 0.8. During Stage II, connectivity measures continue to decrease, but none fall below 0.7. During Stage III, the network recovers slightly, with connectivity values increasing to 0.9 or above. In Stage IV, connectivity is significantly degraded, with many values falling below 0.3. At this point, 16 edges are removed from the network. Central nodes are disconnected from each other, resulting in

large drops in betweenness and closeness centrality. During Stage V, all edges are added back to the network, and connectivity values return to the 0.9 to 1 range. Figure 6 shows the average centrality measures across each stage, and Figure 7 shows the change in average centrality measures.

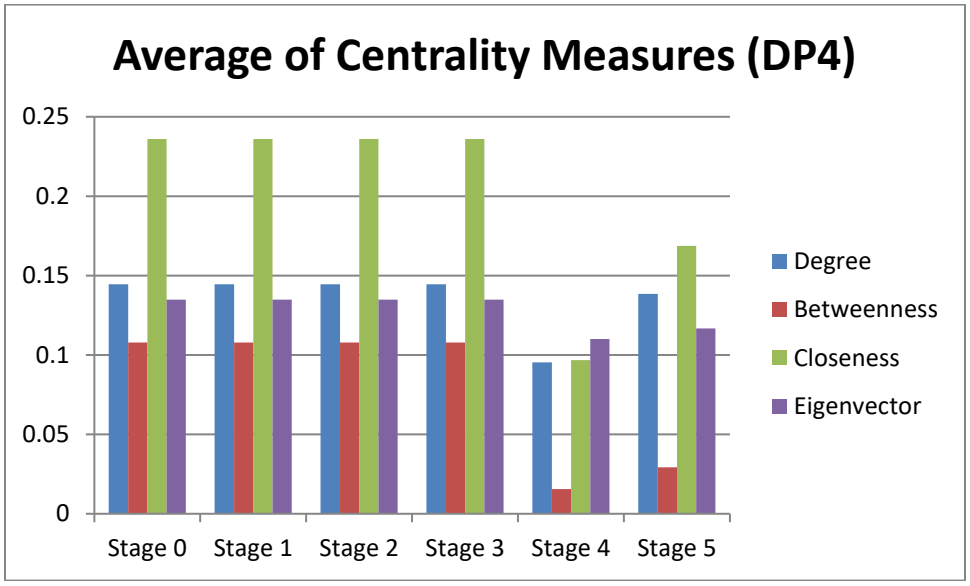


**Figure 6: Design Point 4: Average Centrality Measures.**

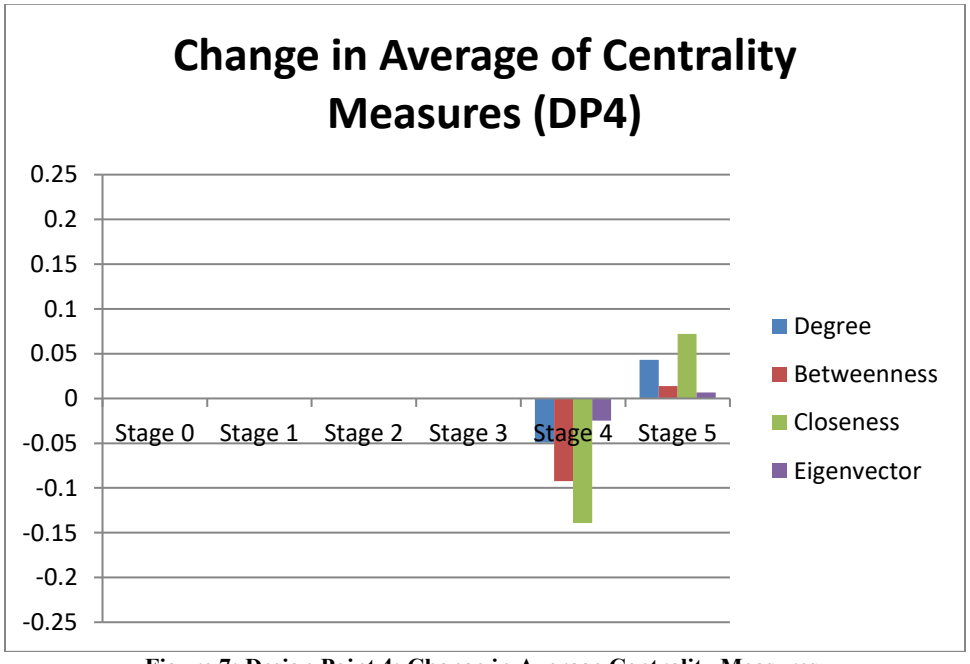


**Figure 7: Design Point 4: Change in Average Centrality Measures.**

**Design Point 5** (attacker: medium; defender: medium) resulted in significant network degradation. I have included all of the connectivity measure graphs for this

design point in Appendix C, which begins on page 39. During Stage I, approximately one quarter of the edges' connectivity measures are reduced to 0.9. In Stage II, thirty-nine edges are removed from the network. In Stage III, and additional three edges are removed. In Stage IV, three more edges are removed from the network. In Stage V, twenty-three edges are added back to the network, and twenty-two edges remain out of the network. Figure 8 shows the average centrality measures for each stage, and Figure 9 shows the changes in average centrality measures between each stage. During Stage II, central nodes are disconnected from the network to the point where the average betweenness centrality drops to zero, indicating that no one node offers access to the rest of the network any more so than all of the other nodes. The network is degraded even more during Stages 3 and 4, and recovers slightly during Stage V.

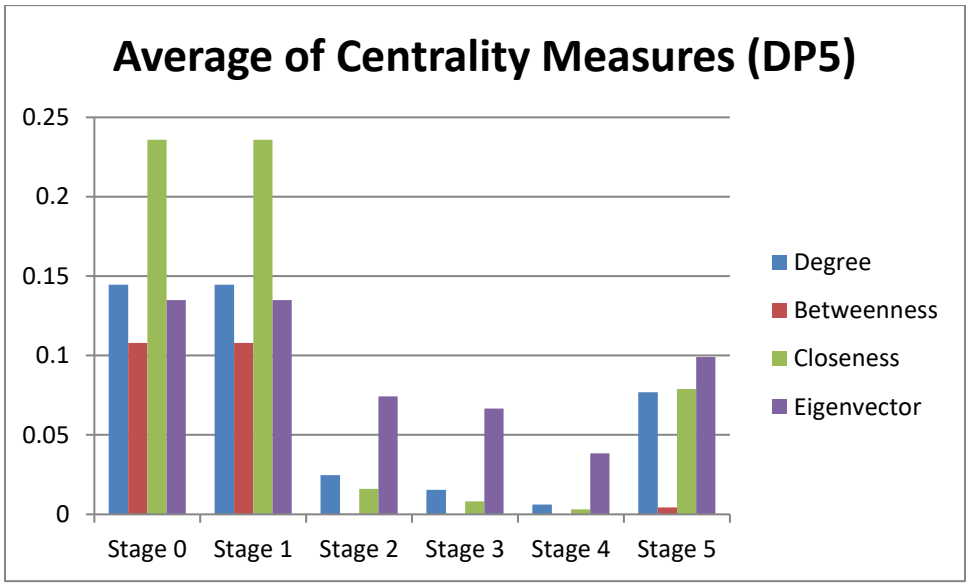


**Figure 8: Design Point 5: Average Centrality Measures.**

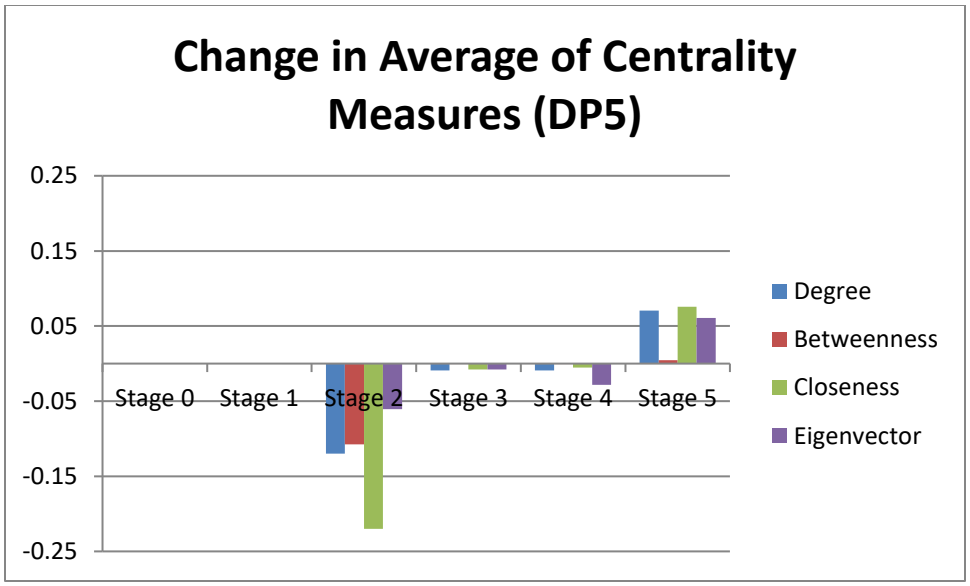


**Figure 9: Design Point 5: Change in Average Centrality Measures.**

Figure 25, located on page 41 in Appendix C, shows the connectivity measures during Stage V, and indicates that all of the values were restored to one. However, the list of edges removed from the network and the average centrality measures indicate that connectivity was not fully restored. This indicates that there is a bug in the simulation program that is prevented edges with restored connectivity from being added back to the network. The graph of average centrality measures for Stage V should resemble that of the initial network (Stage 0).

**Design Point 6** (attacker: medium; defender: high) resulted in minor network connectivity degradation. Connectivity values never fell below 0.7. Figure 10 shows the connectivity measures for each edge during Stage II, when the connectivity measures were most degraded. The graph of average centrality measures for Design Point 6 is the same as that of Design Point 1, above.

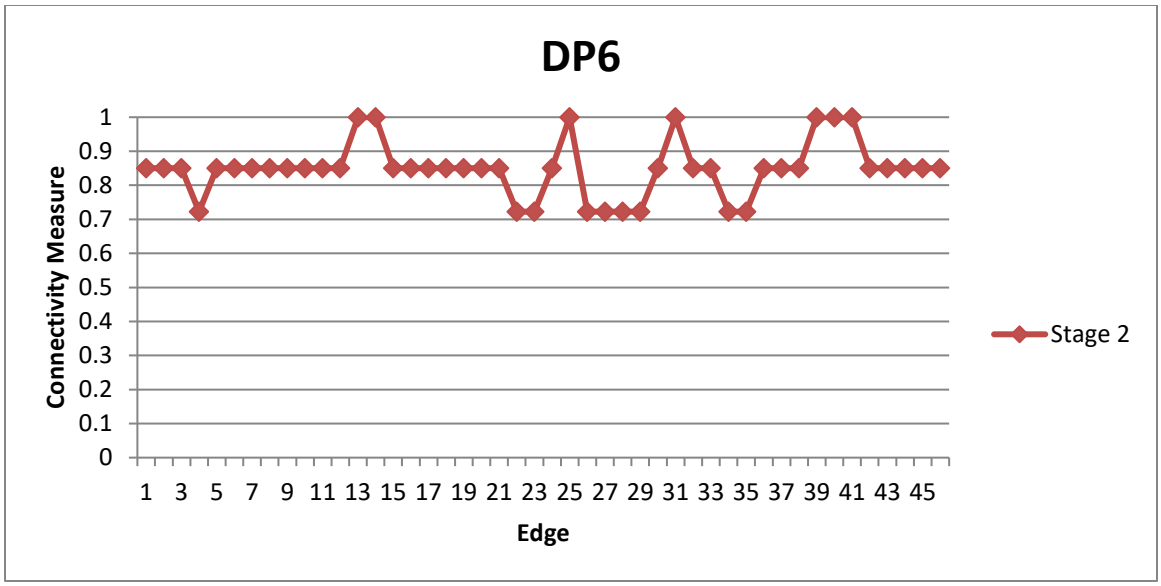


**Figure 10: Design Point 6, Stage II: Connectivity Measures.**

**Design Point 7** (attacker: high; defender: low) resulted in significant network connectivity degradation. I have included all of the connectivity measure graphs for this design point in Appendix D, which begins on page 42. During Stage I, approximately one third of the edges' connectivity measures are reduced to 0.6. During Stages 2 and 3, connectivity values are further degraded, but none fall below the 0.5 threshold. In Stage IV, sixteen edges are removed from the network. In Stage V, fourteen edges are added back to the network, and two edges remain out of the network. Figure 11 shows the average centrality measures for each stage, and Figure 12 shows the changes in average centrality measures between each stage. Central nodes are disconnected during Stage IV, as indicated by the significant decrease in the average betweenness and closeness values. Again, the bug that I discovered in Design Point 5 appeared in Design Point 7. Figure 30, located on page 44 in Appendix D, indicates that all connectivity values fall between 0.9 and 1, but two edges were not added back to the network.

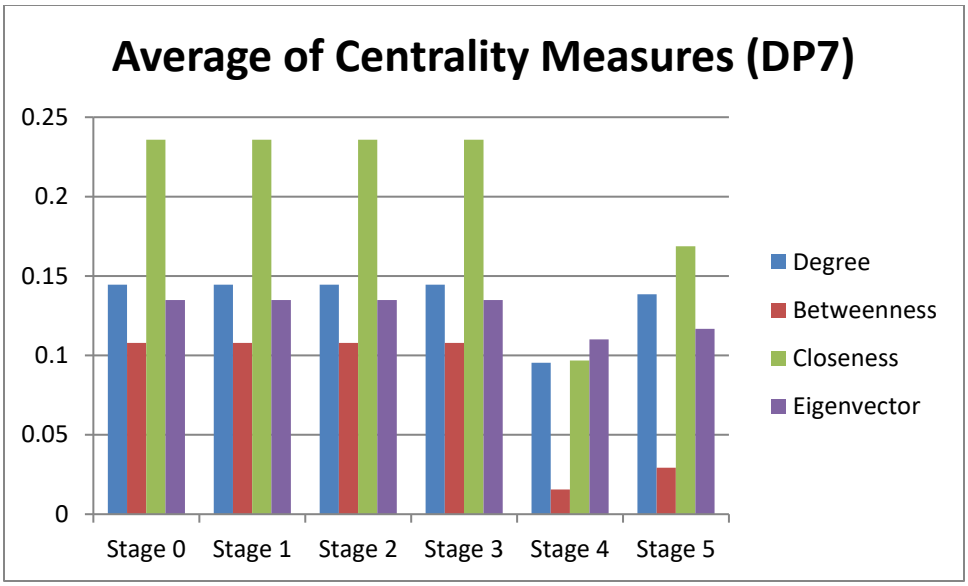


**Figure 11: Design Point 7: Average Centrality Measures.**

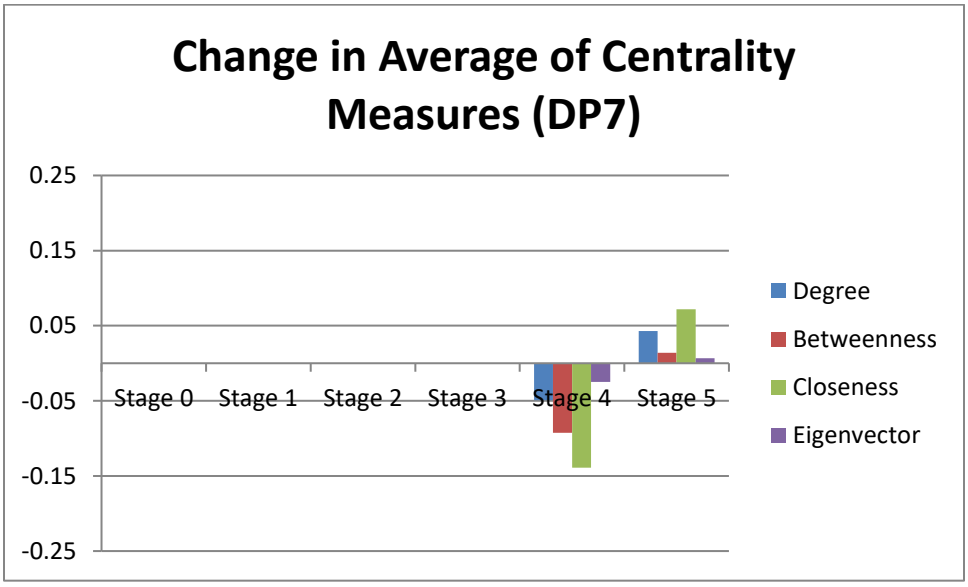


**Figure 12: Design Point 7: Change in Average Centrality Measures.**

**Design Point 8** (attacker: high; defender: medium) resulted in significant network degradation. I have included all of the connectivity measure graphs for this design point in Appendix E, which begins on page 45. During Stage I, approximately one third of the edges' connectivity measures are reduced to 0.7. In Stage II, thirty-nine edges are removed from the network. In Stage III, and additional three edges are removed. In Stage IV, three more edges are removed from the network. In Stage V, twenty-three edges are added back to the network, and twenty-two edges remain out of the network. These results are the same as Design Point 5, and the graphs of average centrality measures and

changes in average centrality measures are also the same (Figure 8 and Figure 9, above). The Stage V bug also appears in this simulation.

**Design Point 9** (attacker: high; defender: high) resulted in minor network connectivity degradation. Connectivity values never fell below 0.6. Figure 13 shows the connectivity measures for each edge during Stage II, when the connectivity measures were most degraded. The graph of average centrality measures for Design Point 9 is the same as that of Design Point 1 (Figure 3, above).

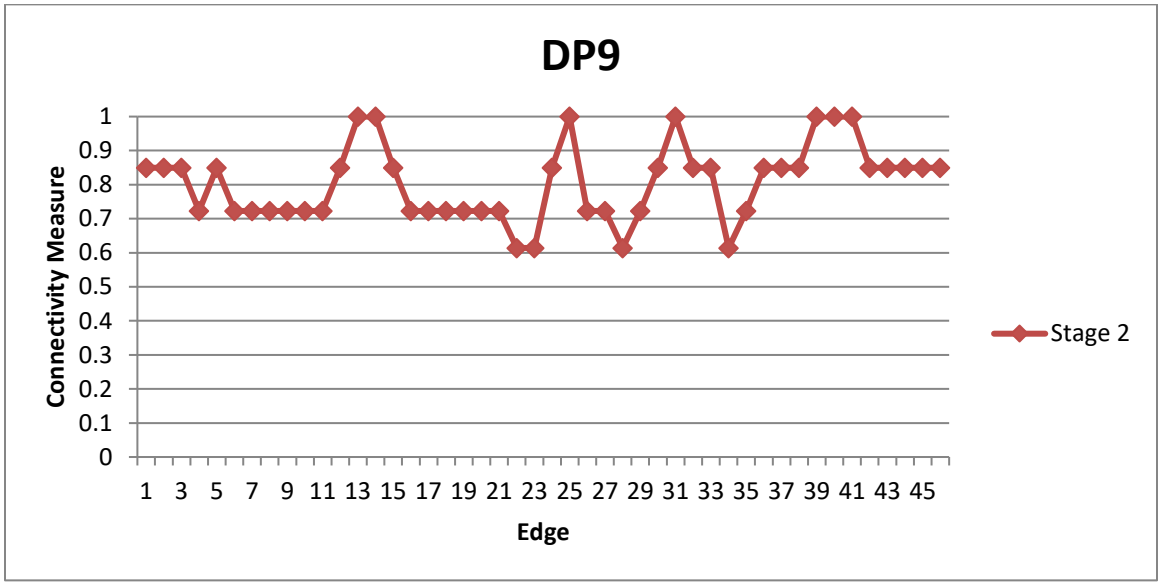


**Figure 13: Design Point 9, Stage II: Connectivity Measures.**

I organized the aggregate data into two types: a full aggregate of all of the design points, and a selected aggregate consisting only of the design points that had edges removed from the network. The aggregate and selected aggregate connectivity measures for each stage can be found in Appendix F, which begins on page 48. The full aggregate data shows the general trend of decreases in connectivity values beginning in Stage I, continuing into Stage II, with a slight recovery in Stage III, followed by a decrease in connectivity in Stage IV and significant recovery in Stage V. The selected aggregate data follows a similar pattern, except there is even less of a recovery during Stage III, and there is significantly more connectivity degradation in Stage IV. The aggregate and selected aggregates of the average centrality measures are located in Figure 14 and Figure 15, below. Both highlight the same general trend, but the effects of removing edges from the network are much more pronounced in the selected aggregate.

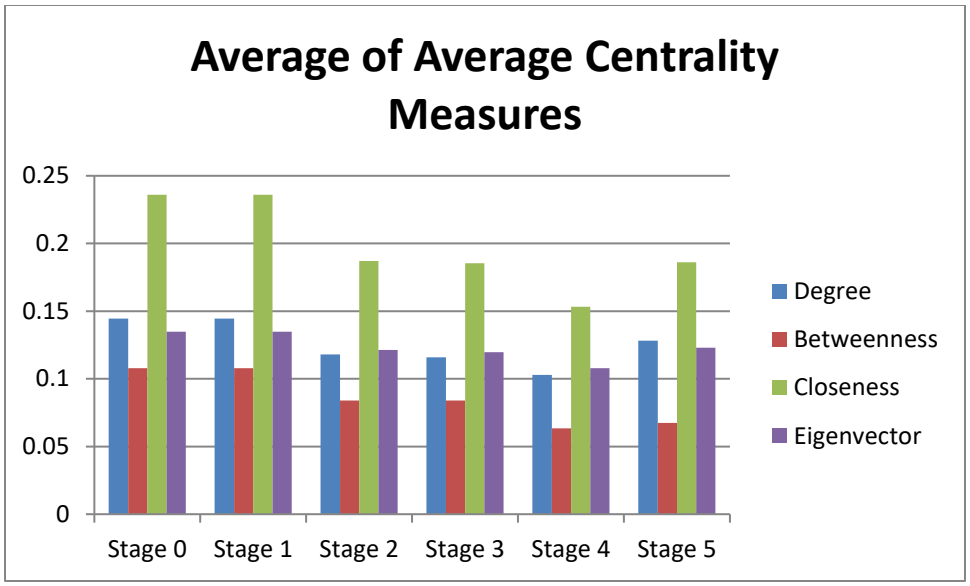


**Figure 14: Aggregate of Average Centrality Measures.**

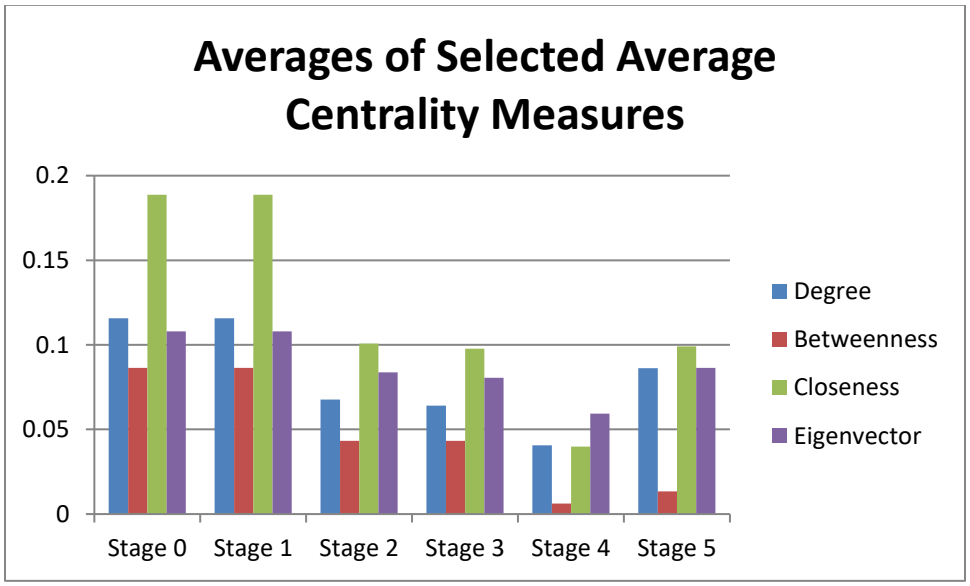


**Figure 15: Selected Aggregate of Average Centrality Measures**

# V. IMPROVEMENTS AND FURTHER RESEARCH

I did not create a counter-attack module during my research. The biggest question that I faced while exploring possible ways to create a counter-attack targeting algorithm exposed a significant limitation in my model and test dataset: the model does not distinguish between attacker- and defender-controlled nodes, and the dataset only includes defender-controlled nodes. Further research should examine whether or not it is realistic to include the attacker-controlled portion of the network. If it is realistic, then my current targeting algorithms would need to be adjusted to differentiate between attacker- and defender-controlled nodes, and the attacks themselves would need to become more specific than my current multiplier system is. Also, the results should report the effects on attacker and defender nodes separately, in order to provide a better understanding of the conflict's effects. If it is not realistic to include the attacker-controlled portion of the network, then I believe the placement of the counter-attack stage in my cyber conflict model should be reevaluated. I believe the best way to model a counter-attack without including attacker nodes would be to degrade the capabilities of the attacker in the next stage. The model currently places the counter-attack stage before Stage V, where I do not believe the effects of the counterattack would have a significant effect. The counter-attack might be better placed before Stage IV, instead. I also did not fully develop the recovery module during my research. Currently, the recovery model has one "standard" setting. Further research should develop my proposed Stage V skill levels for implementation in the simulation.

I noticed one major bug in my program, and several design flaws that could create issues in the future. The bug, which I refer to as the "Stage V Bug," restores all of the connectivity values above the 0.5 threshold for removing edges from the network, but not all of the edges are added back into the network. I suspect that the bug is located in of the "for" loops in the code for the Stage V simulation module, but I have so far been unable to pinpoint the location. Further research should attempt to identify what is causing this situation. The first design flaw is the lack of a failsafe that would prevent edges that have been removed from the network from still being affected when one of the nodes from that edge is attacked. This is because the connectivity measures are stored in a matrix, and the

attack algorithm targets the entire row and column of the target node, and does not skip over values less than 0.5. This flaw has been corrected in a newer version of the CCNS, but is not reflected in these results. This is closely related to a second design flaw, where the simulator does not identify nodes that are isolated from the network or prevent them from being added to targeting lists. This would require writing an algorithm that builds a list of nodes from the deleted edges list that are no longer connected to the network. So far, I have been unable to determine how to automatically detect isolated nodes. A third design flaw is the ability for connectivity values to exceed a value of one. Since the connectivity value is a probability, the values need to be constrained to fall between zero and one. I tried to implement an override that replaces connectivity values that are greater than one with a value of one, but was not successful.

My test of the CCNS maintained the same attacker and defender skill levels across each data point. Further research should examine whether this is a realistic combination of settings for the simulator, since defender's abilities should become degraded after a prolonged attack, or the defender's abilities could become enhanced if skilled cyber warriors were sent to reinforce the network. This research should determine if the simulator should account for fatigue and skill degradation, or if the user should be responsible for incorporating these types of situations by selecting the appropriate settings from the simulator.

The CCNS does not identify critical edges that connect the attacker to the defender, nor does it select targets based on the ability of the attacker to reach those targets, nor does it account for the effects on intermediary nodes and edges when the attacker reaches targets deep within the defender's network. Further research should examine how to identify and protect critical edges in order to ensure the attacker doesn't cripple its own ability to reach the defender's network, and whether or not the target selection algorithm chooses appropriate targets.

Further research should determine how much recovery is appropriate across the various levels of Stage V. Is total recovery realistic? If so, what does the attacker gain if the attacks inflict no lasting damage? Does the motivation for a cyber attack come from the opportunities the attacker is able to exploit while the defenders' networks are down, or does the attacker benefit from the time, resources, and money that the defender spends

on restoring the network? Answering these questions may require the development of different variations of the CCNS to model different types of attackers—military, patriotic hackers, hacktivists, and criminals. One issue in determining the scope of recovery is the lack of a time component within the simulation stages. Total recovery is realistic in the long run, but in the short term, recovery would most likely be only partial. Users may need to define their recovery period in order to determine how much recovery is feasible.

Finally, further research using data from actual cyber conflicts should adjust the skill levels and multipliers I used for the initial test of the CCNS. The current multipliers generate the general trends in network connectivity degradation that I would expect to see, but are not actually supported by other data. If I had actual data, I would fit the attack's timeline to my five-stage model, construct network diagrams for each stage, and determine the skill level of the attacker and defender at each stage. I would then adjust the skill level values and targeting algorithms until the simulator output reasonably matched the network diagrams for the actual cyber conflict. Currently, skill level values are arbitrarily set across all stages. Future work should identify stage-specific skill level values, and the skill levels themselves should be expanded beyond the current three level model in order to provide more realistic options for calculating multipliers. After stage-specific multipliers are developed, the constants should not be necessary, as the values themselves will generate the appropriate multiplier without the need for any adjustments.

# VI. CONCLUSION

My test of the Cyber Conflict Network Simulator validated it against my cyber conflict model. Network degradation occurred during the appropriate stages, and the network recovered where appropriate. When the attacker overmatched the defender, there was more network connectivity degradation, and when the defender overmatched the attacker, there was less. The CCNS is still in its infancy, and requires additional research and testing before its true value can be assessed. Once the CCNS is validated with real-world data, the results will become useful for commanders and policymakers.

Since the CCNS is written in Python, trained users will be able to easily select and customize the simulation's execution parameters and modify algorithms based on specific circumstances and requirements. Commanders and policymakers will be able to tailor the targeting algorithms and other settings in order to identify critical network infrastructure components for different scenarios, and determine and validate ways to introduce redundancies, such as additional hubs and routers, into their networks that mitigate the effects of cyber conflict.

# APPENDIX A

| Stage I | | Attacker Value | Defender Value | Raw Multiplier | Constant | Adjusted Multiplier |
|---|---|---|---|---|---|---|
| Attacker | Defender | | | | | |
| Low | Low | 1 | 1 | 1 | 0 | 0 |
| Low | Medium | 1 | 5 | 5 | 0 | 0 |
| Low | High | 1 | 10 | 10 | 0 | 0 |
| Medium | Low | 5 | 1 | 0.2 | 4 | 0.8 |
| Medium | Medium | 5 | 5 | 1 | 0.9 | 0.9 |
| Medium | High | 5 | 10 | 2 | 0 | 0 |
| High | Low | 10 | 1 | 0.1 | 6 | 0.6 |
| High | Medium | 10 | 5 | 0.5 | 1.4 | 0.7 |
| High | High | 10 | 10 | 1 | 0.85 | 0.85 |

**Table 5: Stage I Skill Levels, Skill Values, Constants, and Multipliers.**

| Stage II | | Attacker Value | Defender Value | Raw Multiplier | Constant | Adjusted Multiplier |
|---|---|---|---|---|---|---|
| Attacker | Defender | | | | | |
| Low | Low | 1 | 1 | 1 | 0.85 | 0.85 |
| Low | Medium | 1 | 5 | 5 | 0.18 | 0.9 |
| Low | High | 1 | 10 | 10 | 0.095 | 0.95 |
| High | Low | 10 | 1 | 0.1 | 9.5 | 0.95 |
| High | Medium | 10 | 5 | 0.5 | 1 | 0.5 |
| High | High | 10 | 10 | 1 | 0.85 | 0.85 |

**Table 6: Stage II Skill Levels, Skill Values, Constants, and Multipliers.**

| Stage III | | Attacker Value | Defender Value | Raw Multiplier | Constant | Adjusted Multiplier |
|---|---|---|---|---|---|---|
| Attacker | Defender | | | | | |
| Low | Low | 1 | 1 | 1 | 1.5 | 1.5 |
| Low | High | 1 | 10 | 10 | 0.195 | 1.95 |
| Medium | Low | 5 | 1 | 0.2 | 6.25 | 1.25 |
| Medium | High | 5 | 10 | 2 | 0.825 | 1.65 |
| High | Low | 10 | 1 | 0.1 | 10.5 | 1.05 |
| High | High | 10 | 10 | 1 | 1.25 | 1.25 |

**Table 7: Stage III Skill Levels, Skill Values, Constants, and Multipliers.**

| Stage IV | | Attacker Value | Defender Value | Raw Multiplier | Constant | Adjusted Multiplier |
|---|---|---|---|---|---|---|
| Attacker | Defender | | | | | |
| Low | Low | 1 | 1 | 1 | 0.85 | 0.85 |
| Low | Medium | 1 | 5 | 5 | 0.18 | 0.9 |
| Low | High | 1 | 10 | 10 | 0.095 | 0.95 |
| High | Low | 10 | 1 | 0.1 | 2.5 | 0.25 |
| High | Medium | 10 | 5 | 0.5 | 1 | 0.5 |
| High | High | 10 | 10 | 1 | 0.85 | 0.85 |

**Table 8: Stage IV Skill Levels, Skill Values, Constants, and Multipliers.**

| Stage V<br>All | Attacker Value | Defender Value | Raw Multiplier | Constant | Adjusted Multiplier |
|---|---|---|---|---|---|
| Standard | 5 | 10 | 2 | 1 | 2 |

**Table 9: Stage V Skill Levels, Skill Values, Constants, and Multipliers.**

# APPENDIX B

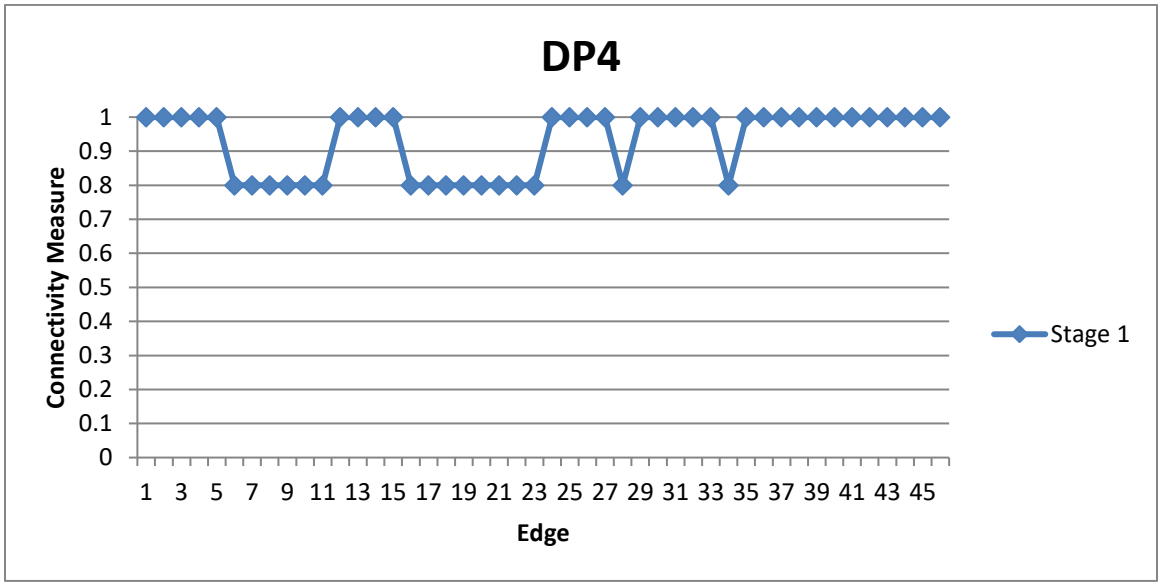


**Figure 16: Design Point 4, Stage I: Connectivity Measures.**

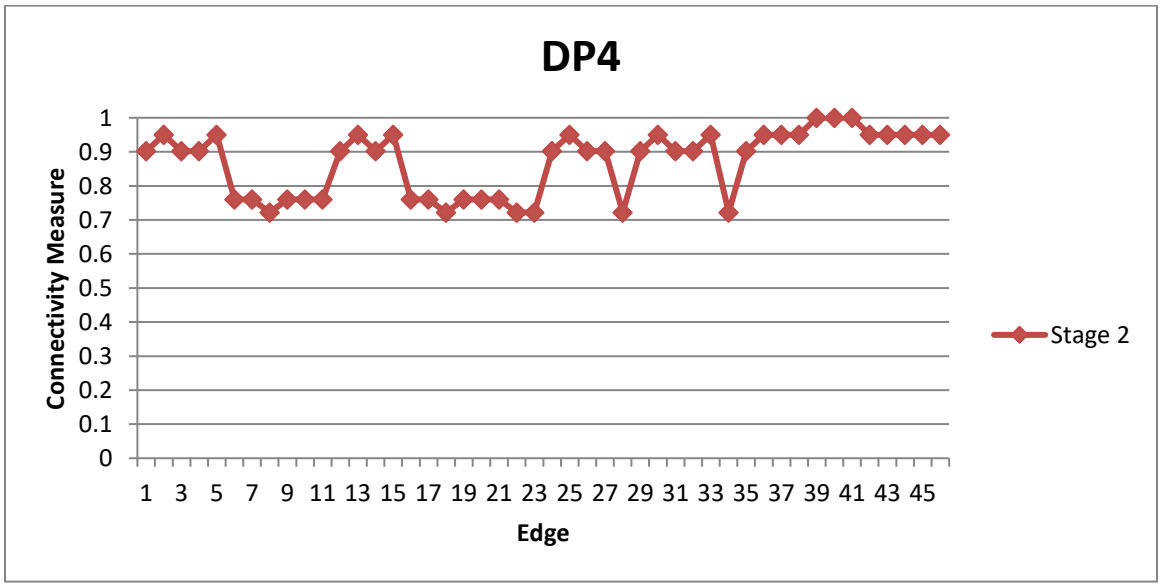


**Figure 17: Design Point 4, Stage II: Connectivity Measures.**

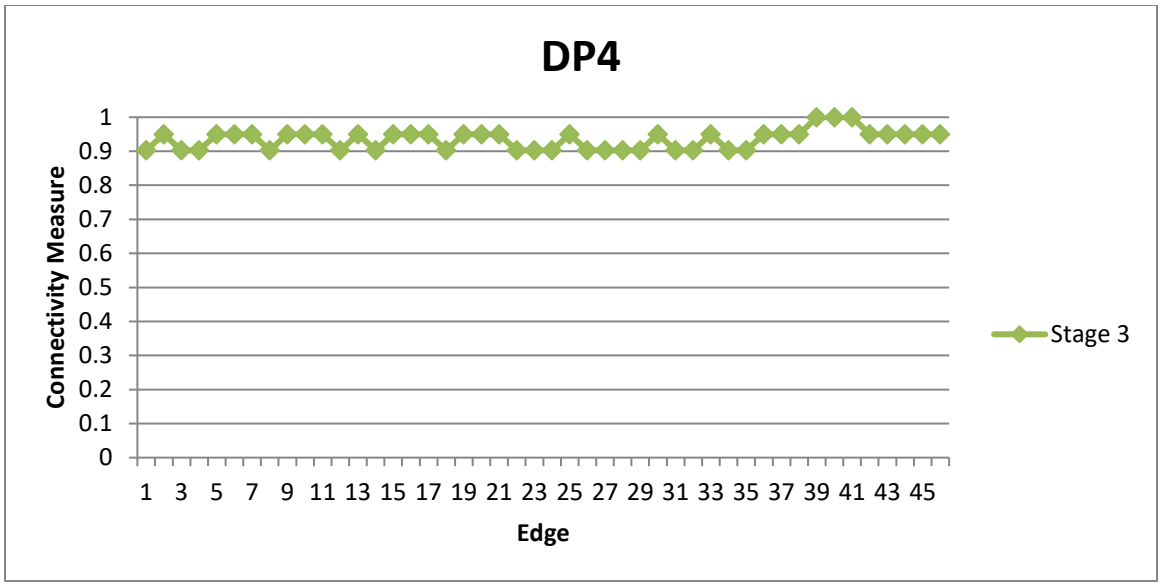


**Figure 18: Design Point 4, Stage III: Connectivity Measures.**

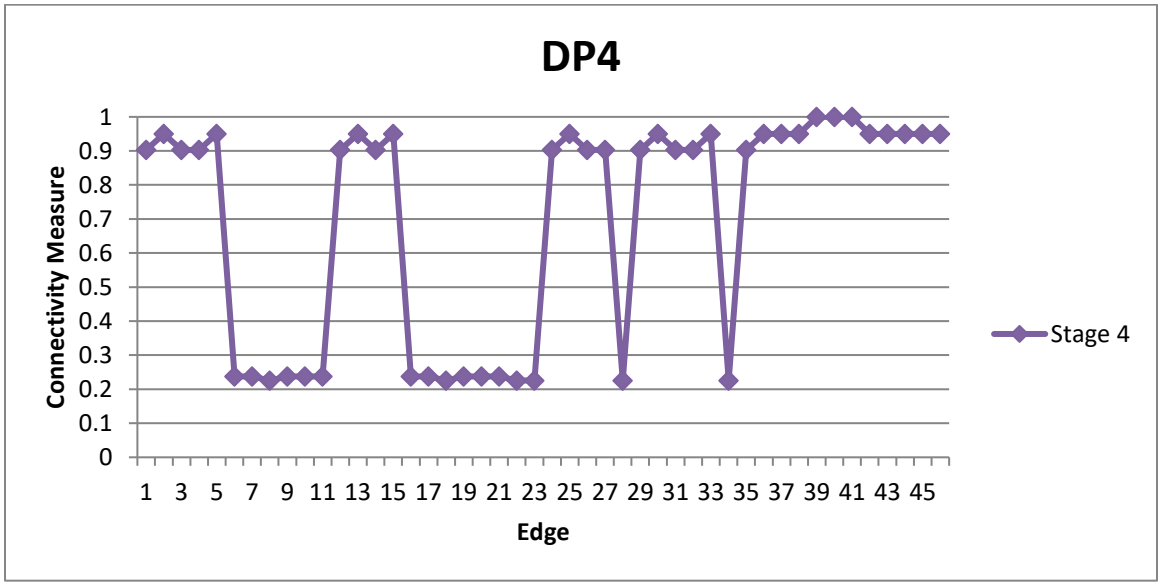


**Figure 19: Design Point 4, Stage IV: Connectivity Measures.**

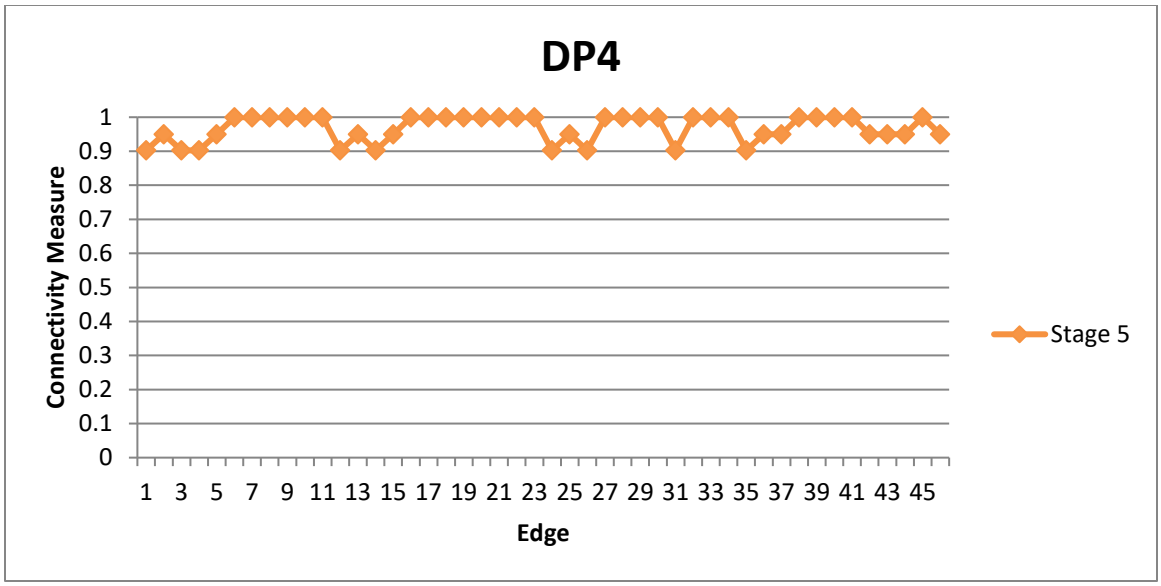


**Figure 20: Design Point 4, Stage V: Connectivity Measures.**

# APPENDIX C

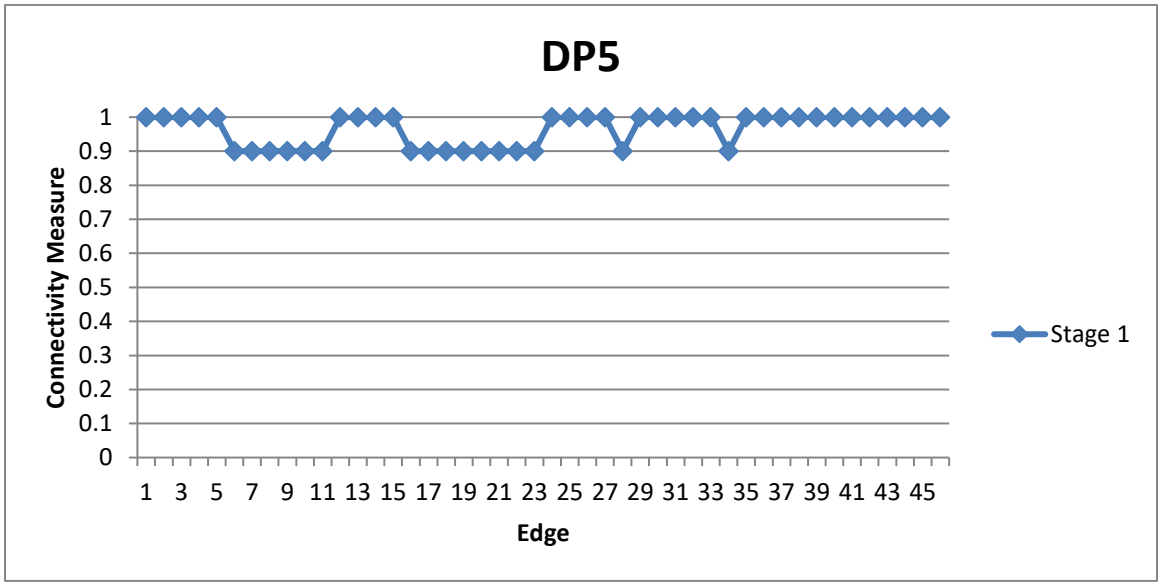


**Figure 21: Design Point 5, Stage I: Connectivity Measures.**

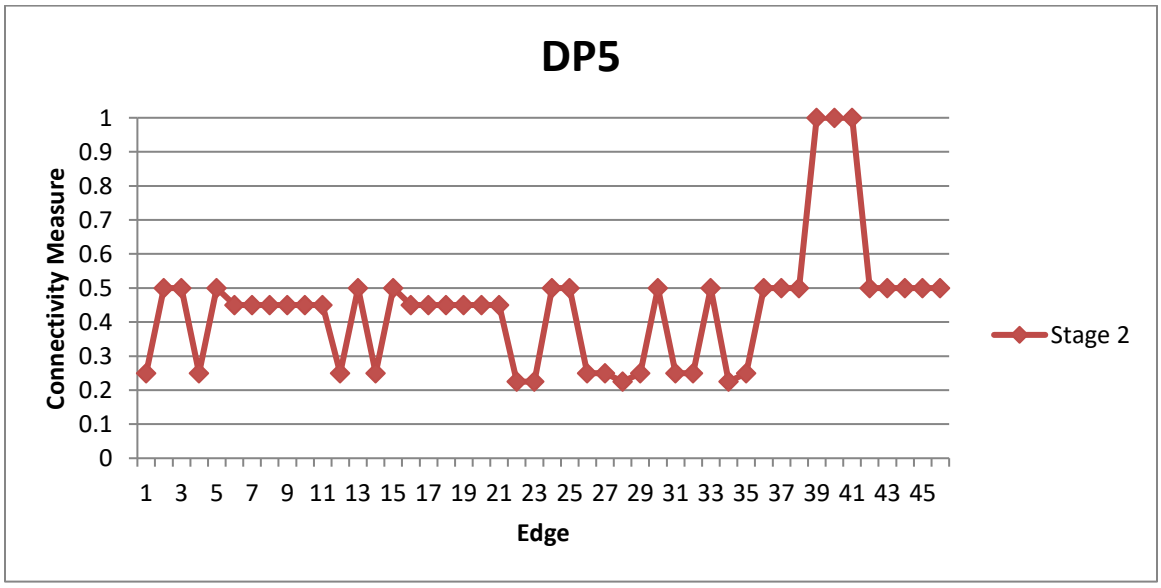


**Figure 22: Design Point 5, Stage II: Connectivity Measures.**

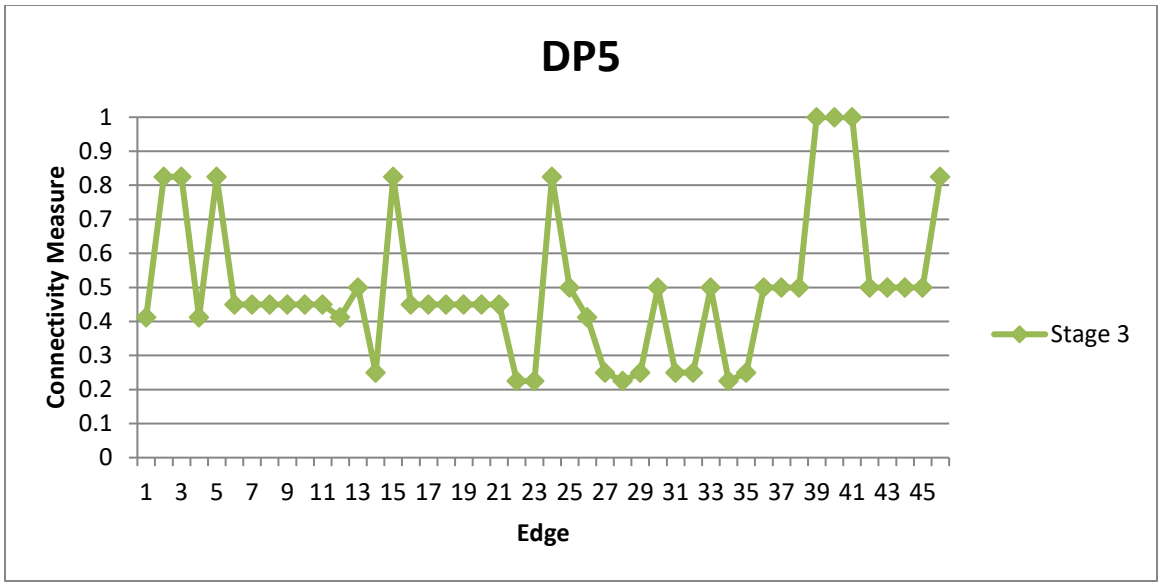


**Figure 23: Design Point 5, Stage III: Connectivity Measures.**

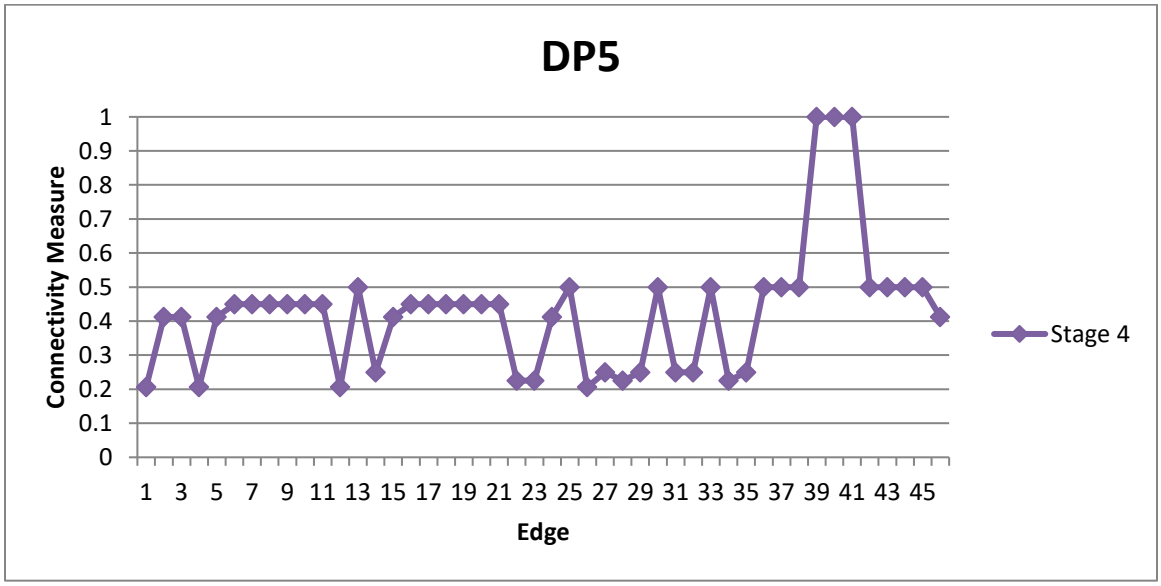


**Figure 24: Design Point 5, Stage IV: Connectivity Measures.**

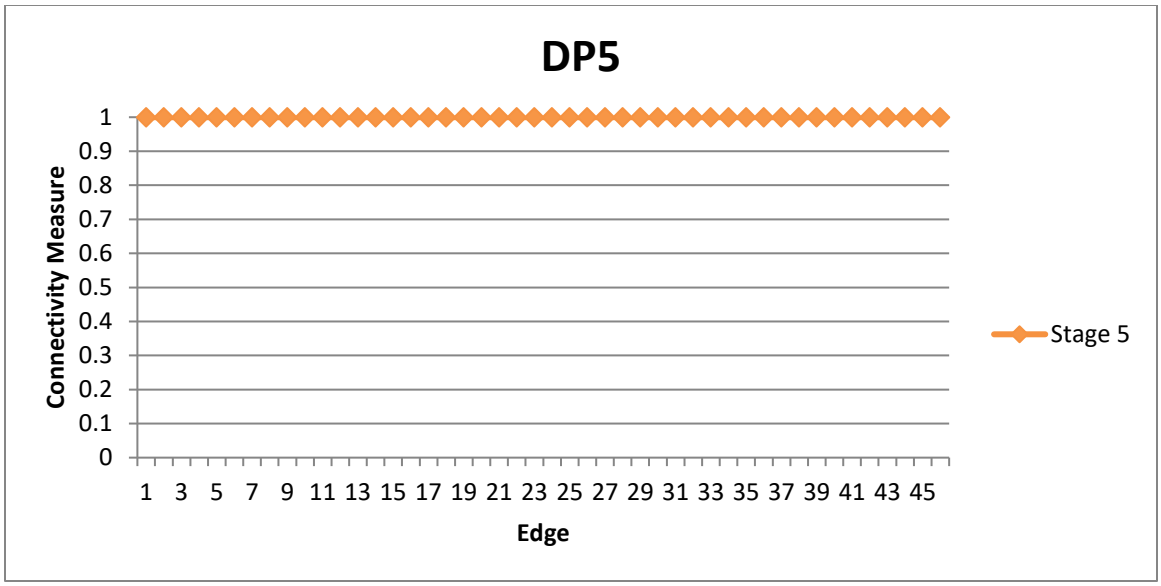


**Figure 25: Design Point 5, Stage V: Connectivity Measures.**

# APPENDIX D

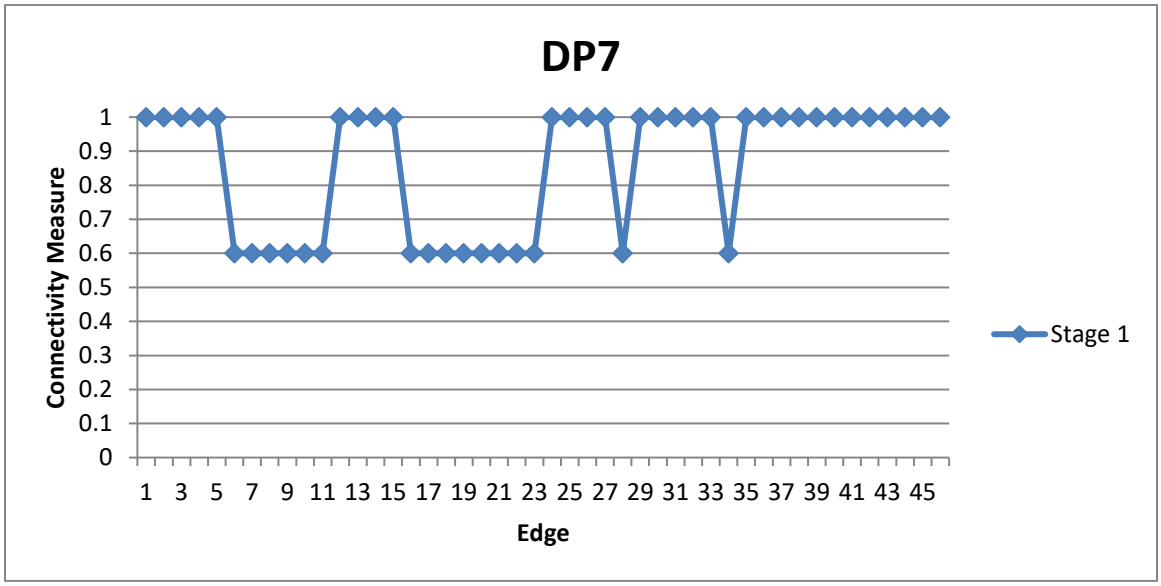


**Figure 26: Design Point 7, Stage I: Connectivity Measures.**

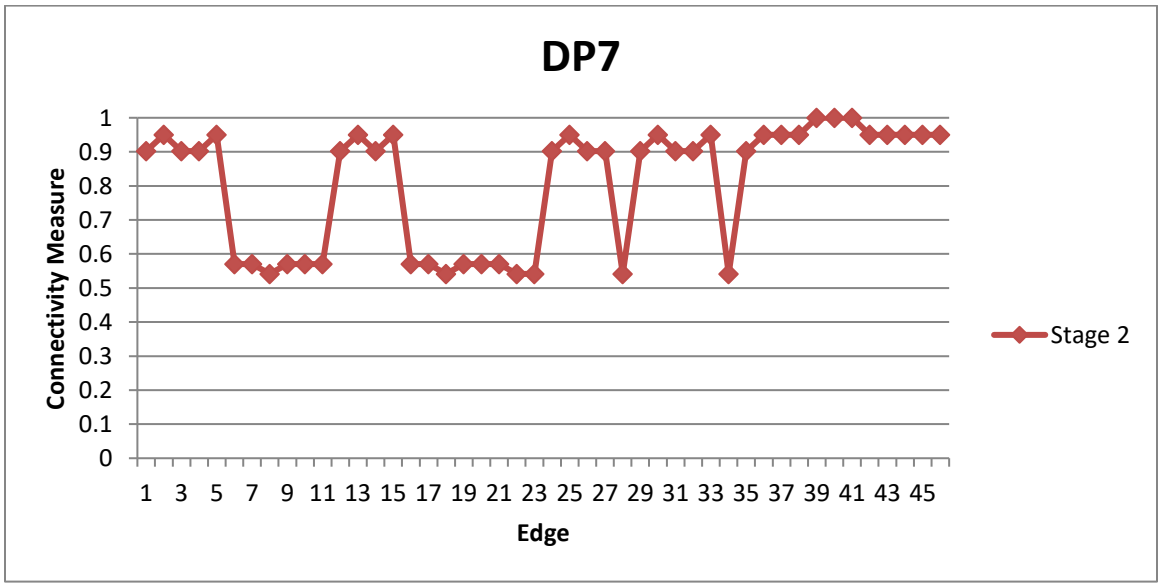


**Figure 27: Design Point 7, Stage II: Connectivity Measures.**

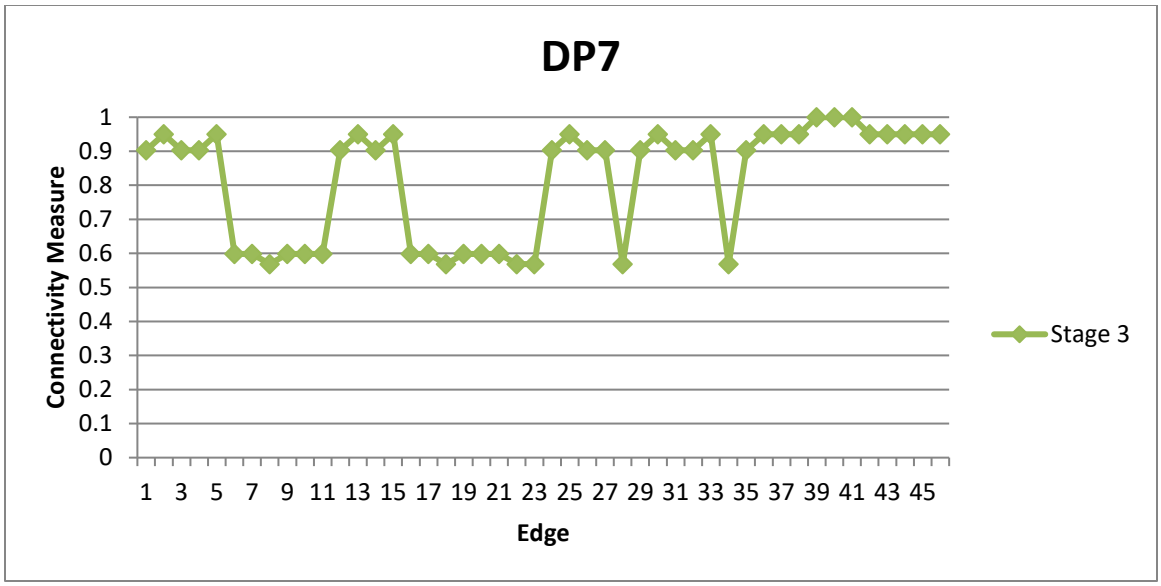


**Figure 28: Design Point 7, Stage III: Connectivity Measures.**

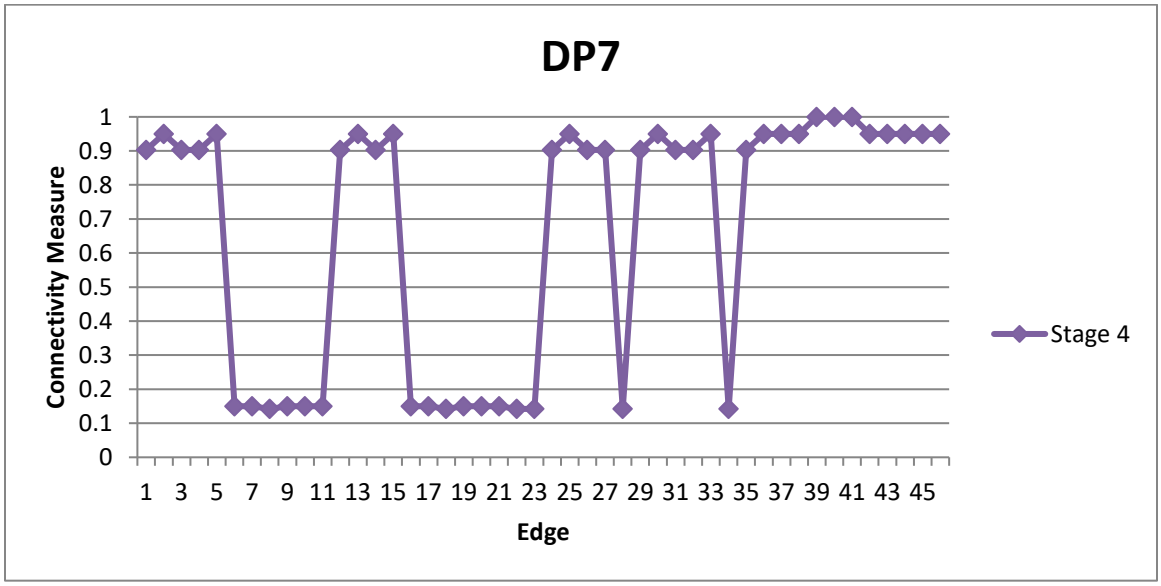


**Figure 29: Design Point 7, Stage IV: Connectivity Measures.**

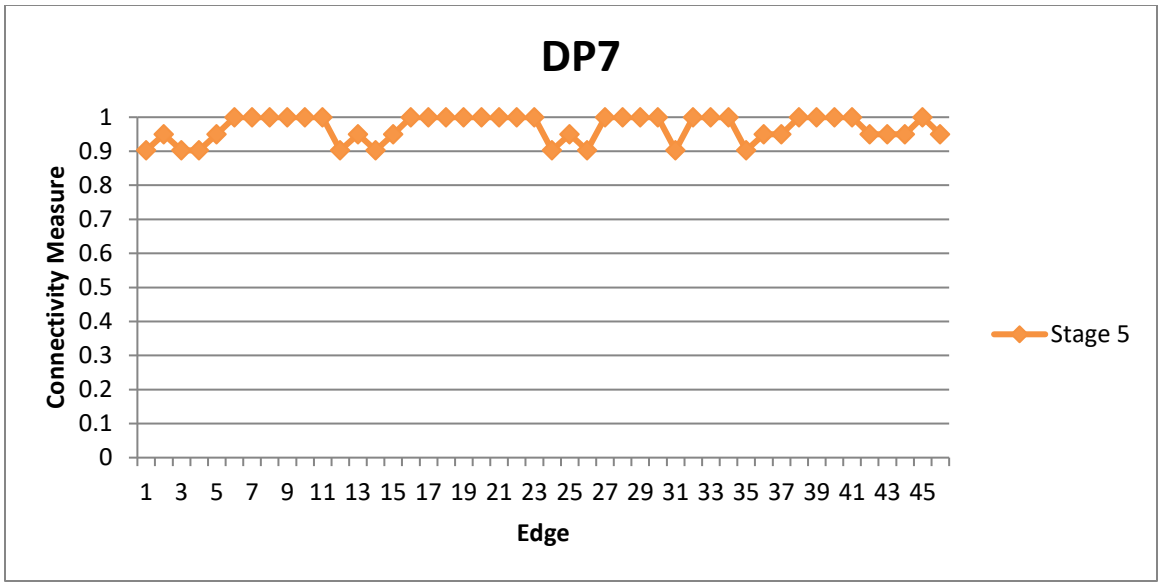


**Figure 30: Design Point 7, Stage V: Connectivity Measures.**

# APPENDIX E

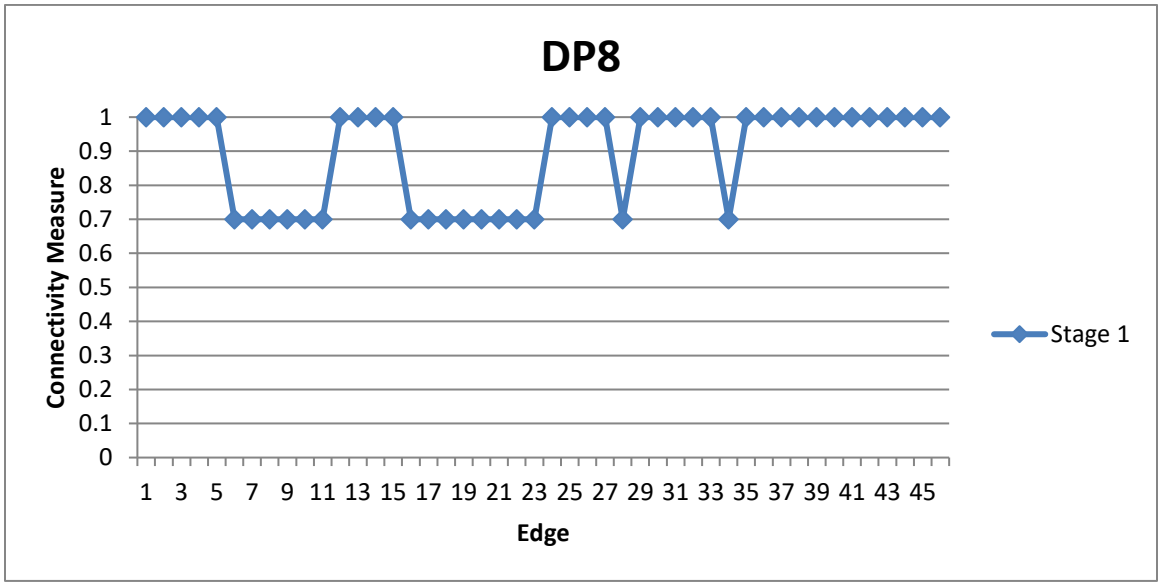


**Figure 31: Design Point 8, Stage I: Connectivity Measures.**

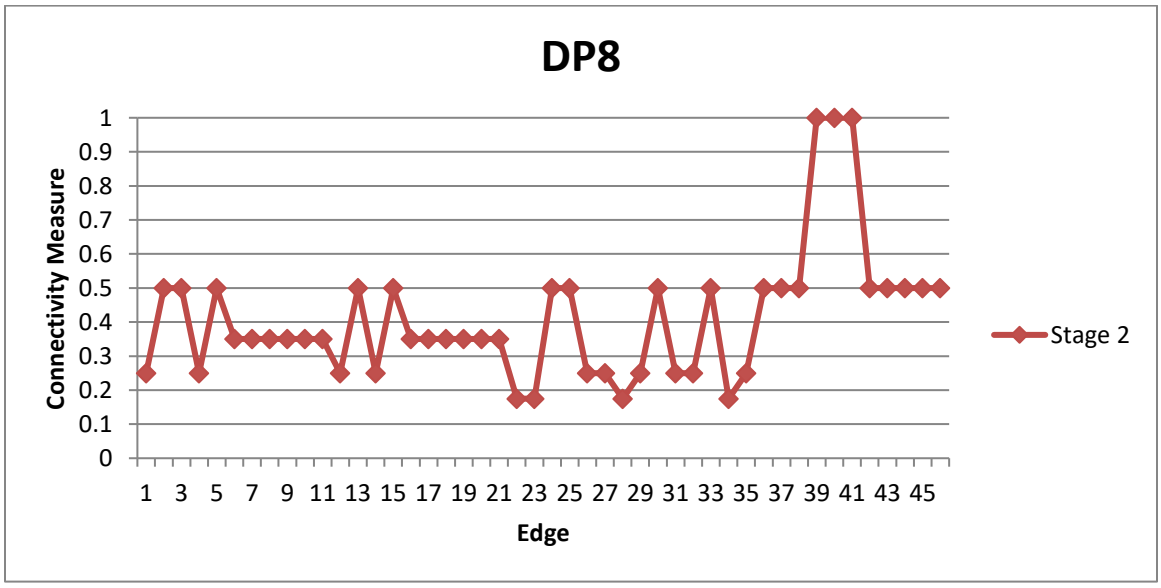


**Figure 32: Design Point 8, Stage II: Connectivity Measures.**

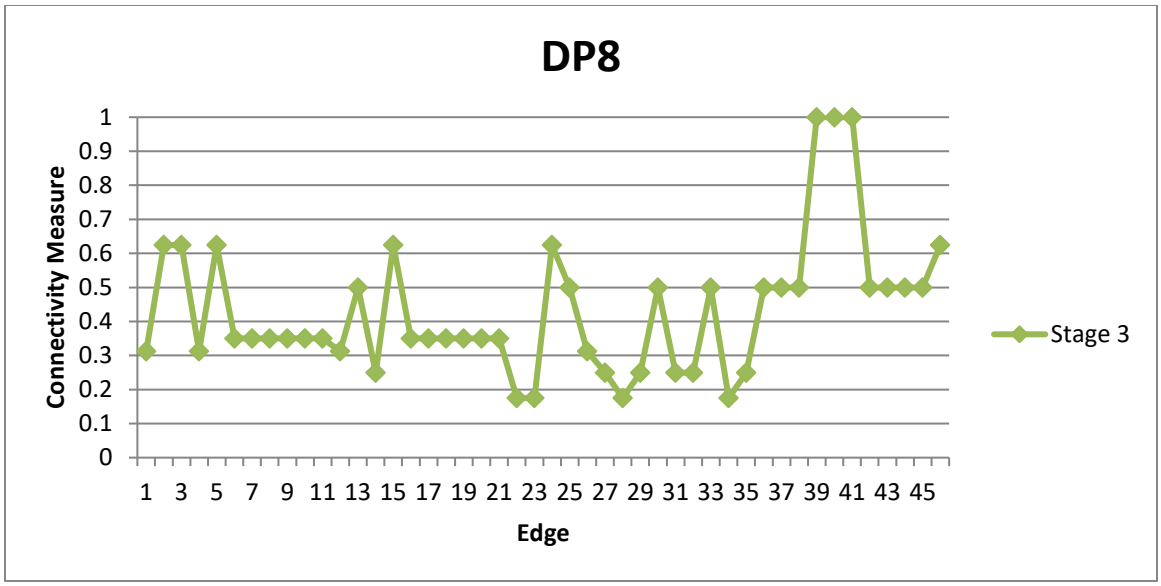


**Figure 33: Design Point 8, Stage III: Connectivity Measures.**

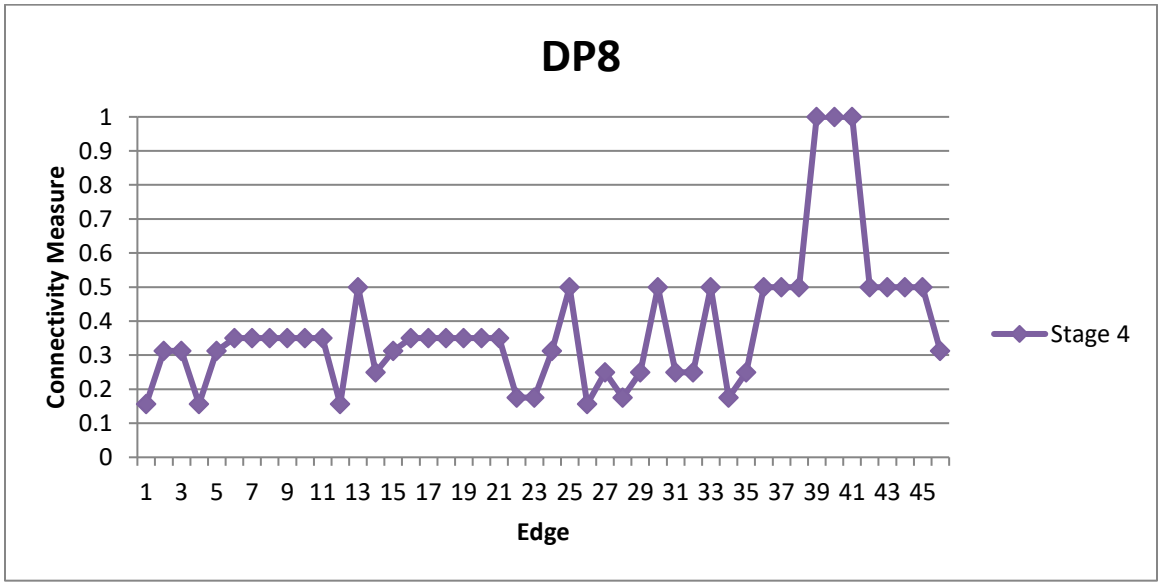


**Figure 34: Design Point 8, Stage IV: Connectivity Measures.**

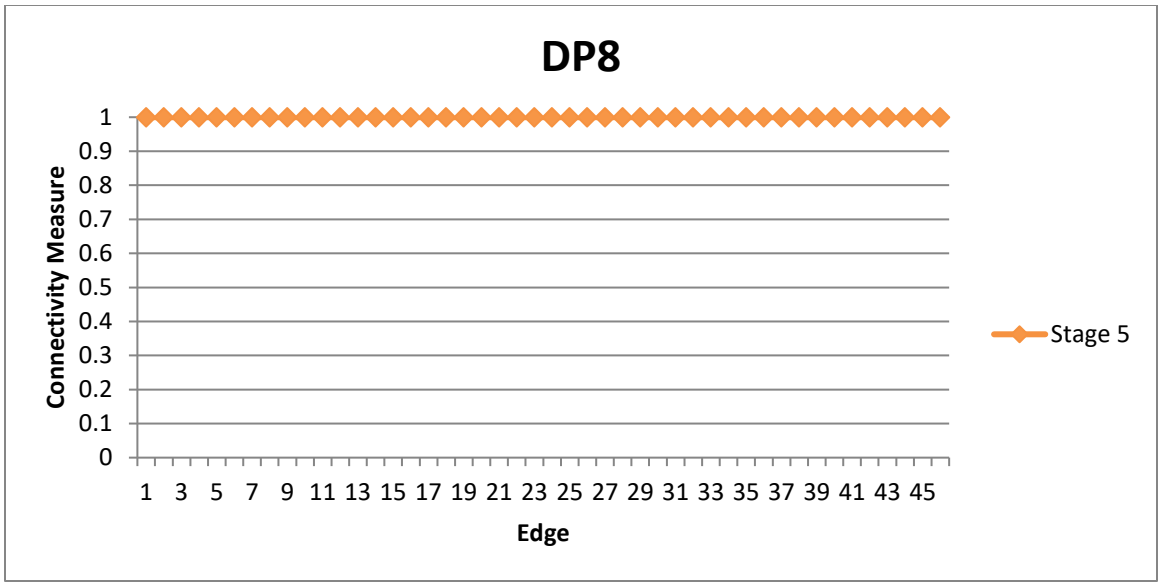


**Figure 35: Design Point 8, Stage V: Connectivity Measures.**

# APPENDIX F

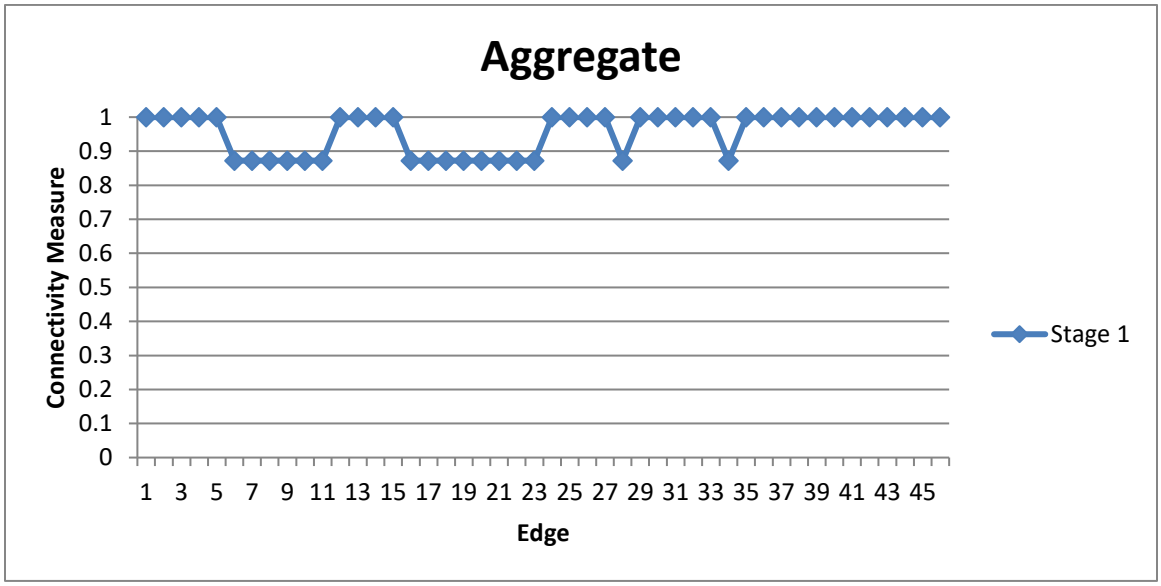


**Figure 36: Aggregate Data, Stage I: Connectivity Measures.**

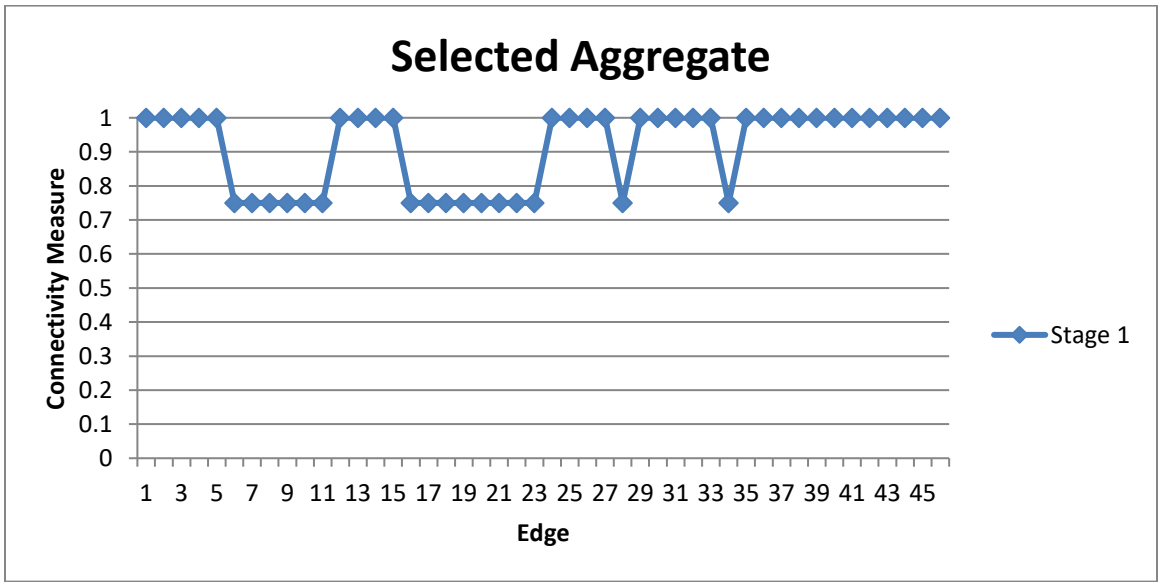


**Figure 37: Selected Aggregate Data, Stage I: Connectivity Measures.**

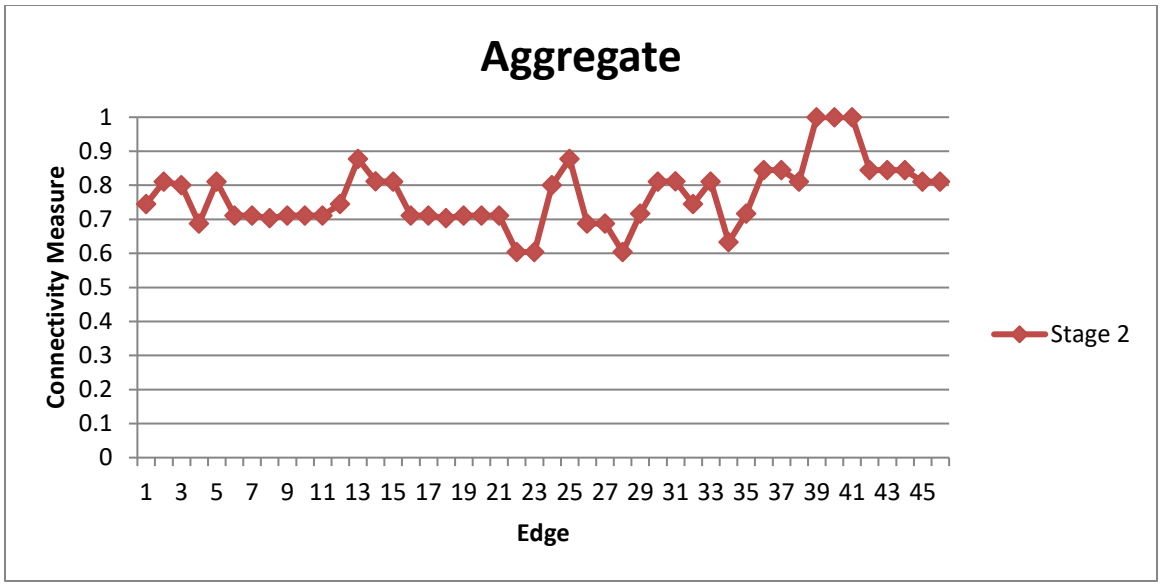


**Figure 38: Aggregate Data, Stage II: Connectivity Measures.**

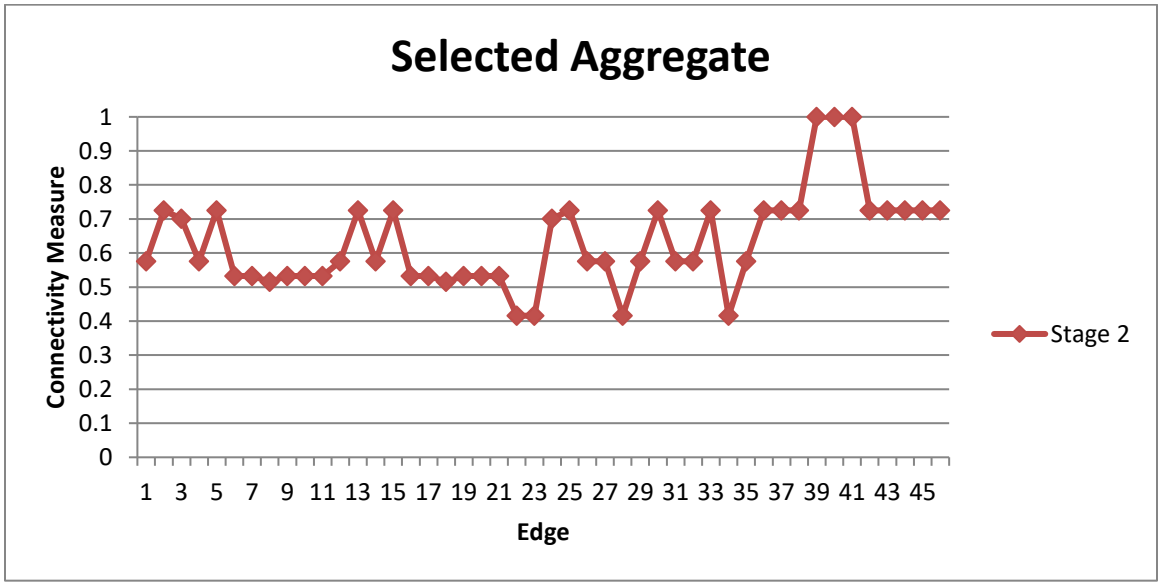


**Figure 39: Selected Aggregate Data, Stage II: Connectivity Measures.**

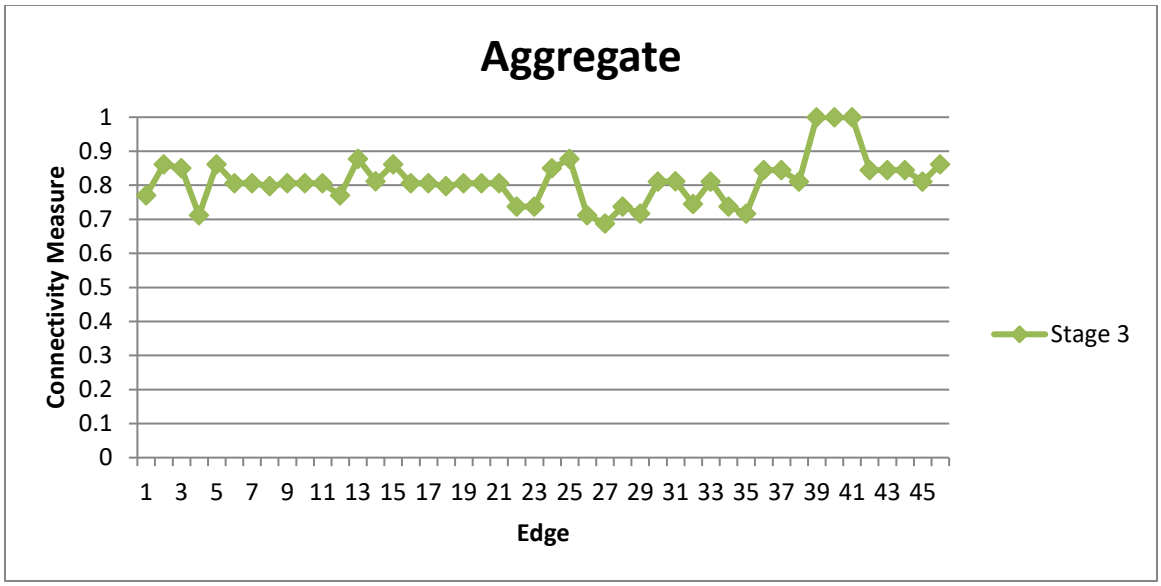


**Figure 40: Aggregate Data, Stage III: Connectivity Measures.**

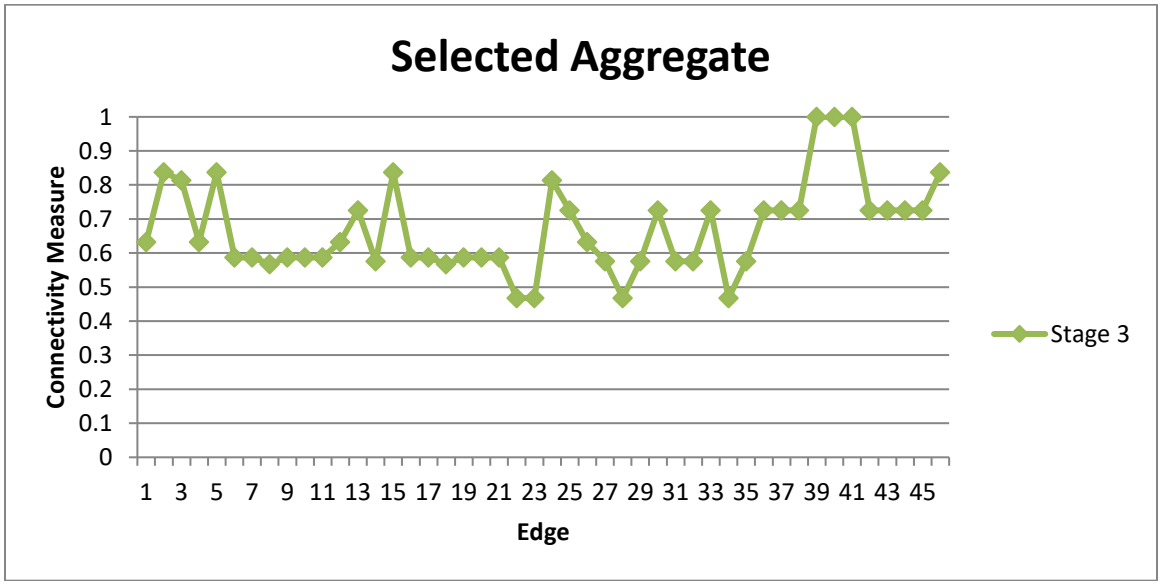


**Figure 41: Selected Aggregate Data, Stage III: Connectivity Measures.**

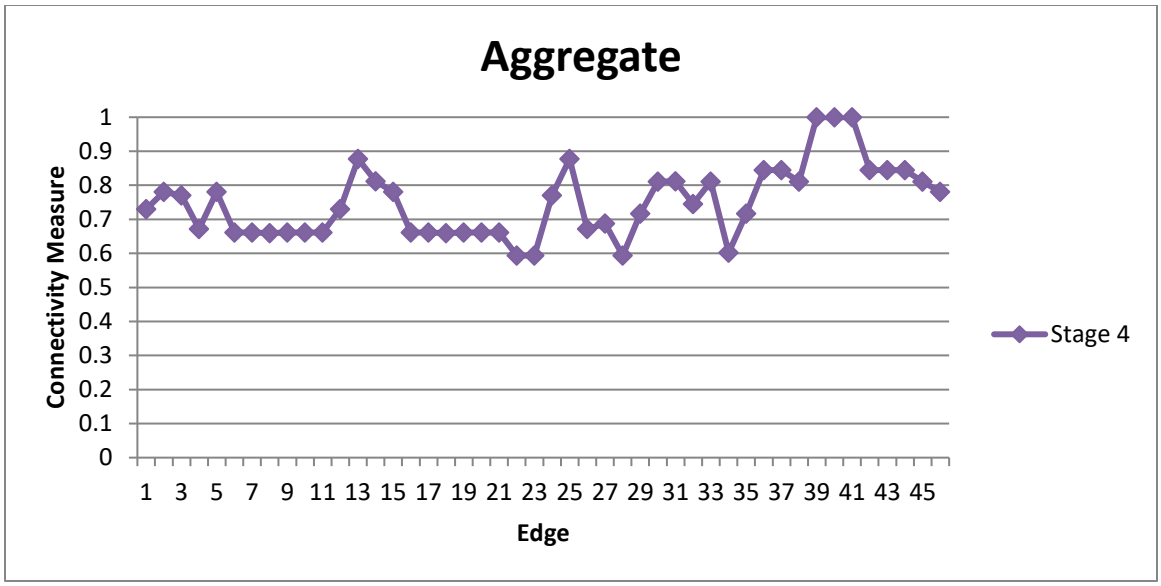


**Figure 42: Aggregate Data, Stage IV: Connectivity Measures.**

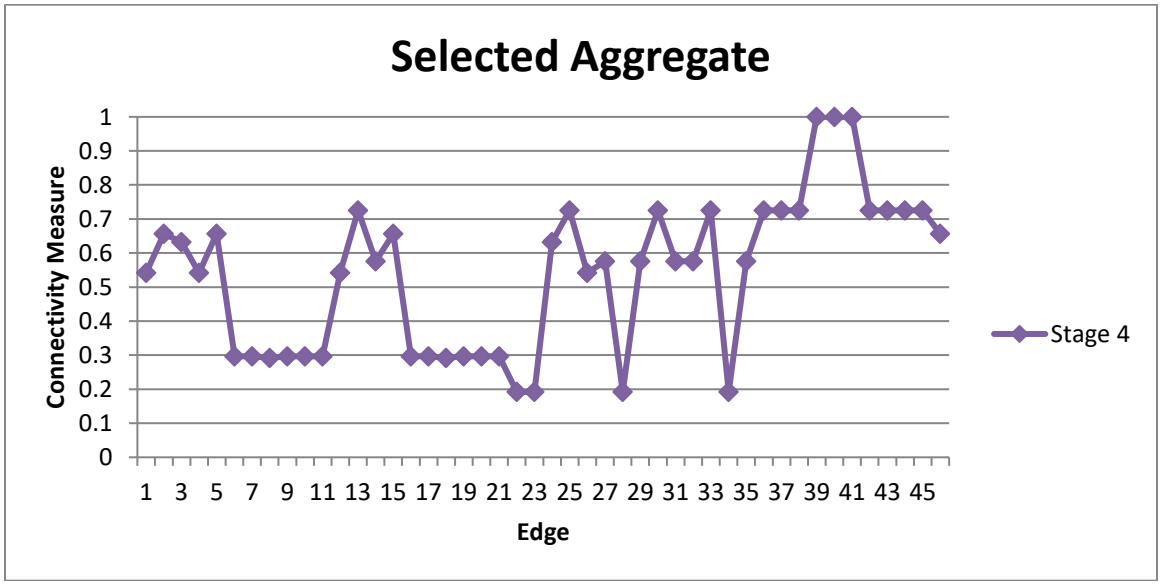


**Figure 43: Selected Aggregate Data, Stage IV: Connectivity Measures.**

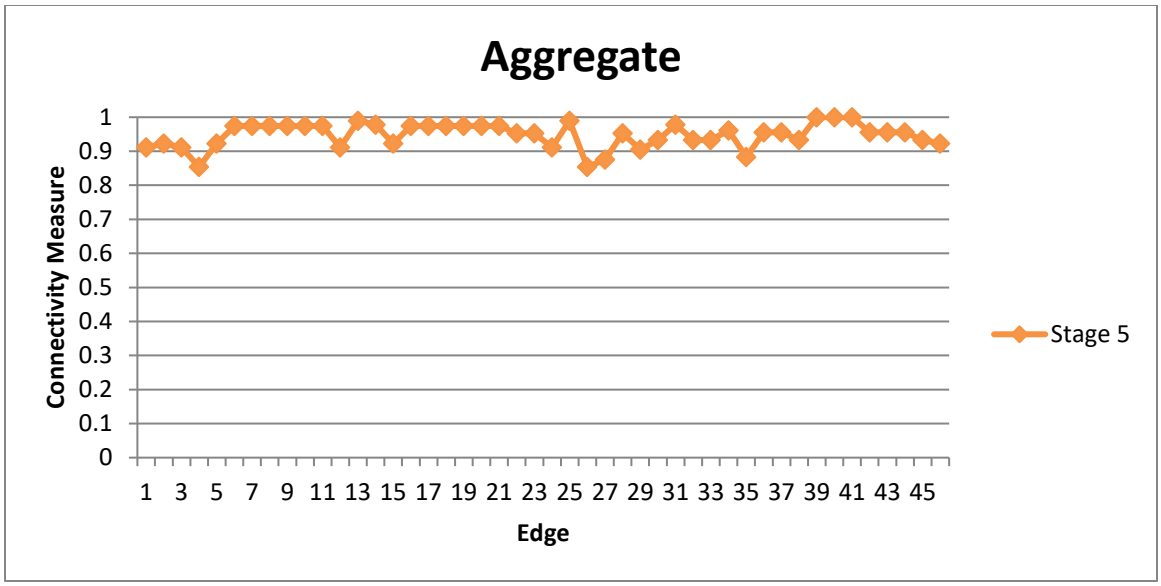


**Figure 44: Aggregate Data, Stage V: Connectivity Measures.**

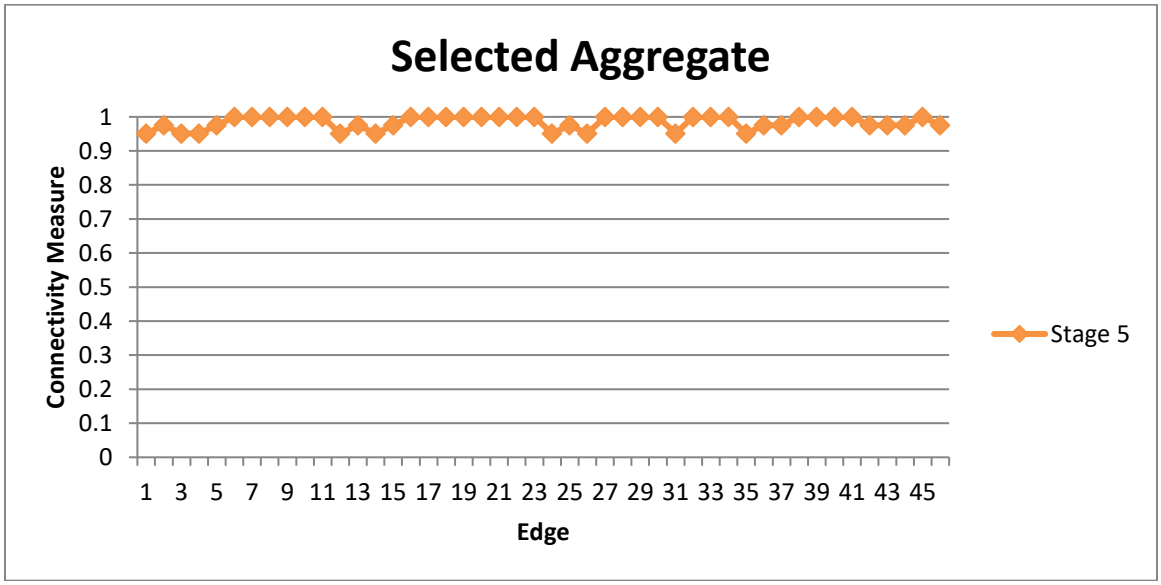


**Figure 45: Selected Aggregate Data, Stage V: Connectivity Measures.**

# LIST OF REFERENCES


Barzashka, I. (2013, April 28). Are cyber-weapons effective? *The RUSI Journal*, 158(2). doi:10.1080/03071847.2013.787735.

BBC. (2013, December 19). NSA: White House task force recommends surveillance curbs. Retrieved from http://www.bbc.co.uk/news/world-us-canada-25439544.

Clark, R. A., & Knake, R. K. (2010). *Cyber war: The next threat to national security and what to do about it*. New York, NY: HarperCollins.

Committee on Network Science for Future Army Applications. (2005). *Network science*. Retrieved from http://site.ebrary.com/lib/usma/.

Cdx 2009 network Usma. (2009, April 02). Retrieved from http://www.westpoint.edu/crc/SiteAssets/SitePages/DataSets/CDX_2009_Network_USMA.pdf .

Engebretson, P. (2011). *The basics of hacking and penetration testing: Ethical hacking and penetration testing made easy*. Waltham, MA: Syngress.

Gray, C. S. (2013). *Making strategic sense of cyber power: Why the sky is not falling*. Carlisle Barracks, PA: Army War College.

Healey, J. (Ed.). (2013). *A fierce domain: Conflict in cyberspace, 1986 to 2012* [Kindle version]. Retrieved from Amazon.com.

Johnson, A., McCulloh, I., Curwin, E., & Topp, S. (2013). Advanced network analysis and targeting report (Version 2.0) [Computer software]. West Point, NY: United States Military Academy.

Leiner, B. M., Cerf, C. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., Postel, J., Roberts, L. G., & Wolff, S. (n.d.). *Brief history of the Internet*. Retrieved from http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet.

Lewis, T. G. (2009). *Network science: Theory and applications*. Retrieved from http://site.ebrary.com/lib/usma/.

McCulloh, I., & Johnson, A. (2011). *Advanced network analysis and targeting course: A social network approach to targeting*. Wiley Preprint.

Spafford, E. H. (1988, December 8). The internet worm program: An analysis. West Lafayette, IN: Purdue. Retrieved from http://spaf.cerias.purdue.edu/tech-reps/823.pdf.