6-8-2009

An Efficient and Effective Implementation of the Trust System for Power Grid Compartmentalization

Juan M. Carlos Gonzalez

An Efficient and Effective Implementation of the Trust System for Power Grid Compartmentalization

THESIS

Juan M. Carlos Gonzalez, Captain, USAF

AFIT/GCS/ENG/09-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GCS/ENG/09-01

An Efficient and Effective Implementation of the Trust System for Power Grid Compartmentalization

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science

Juan M. Carlos Gonzalez, B.S.

Captain, USAF

June 2009

AFIT/GCS/ENG/09-01

An Efficient and Effective Implementation of the Trust System for Power Grid Compartmentalization

Juan M. Carlos Gonzalez, B.S.
Captain, USAF

Approved:

/signed/	8 Jun 2009
Dr. Kenneth M. Hopkinson (Chairman)	date
/signed/	8 Jun 2009
Lt Col Stuart J. Kurkowski, PhD (Member)	date
/signed/	8 Jun 2009
Maj Ryan W. Thomas, PhD (Member)	date

Abstract

As utility companies develop and incorporate new technologies, such as moving to utility Internet technology based architecture and standard; it is crucial that we do so with history in mind. We know that traditional utility protection and control systems were not designed with security in their top priorities. This presents a danger in an environment where near real-time responses are required to ensure safe operations. As a consequence, system security becomes a burden to the system rather than necessary protection.

Unfortunately, technology implementation is not the only concern. The number of utility privately-owned companies has multiplied as the market has moved to a deregulated market in an effort to fragment the traditional monopolies that ruled the industry. Additionally, as new technologies replace or are coupled with legacy systems; new risks to our national assets are incorporated. We have to keep in mind recent events that have proven that our nation is vulnerable to attacks that could severely hinder our economy. Unfortunately, this is easier said than done. The conditions under which these systems operate make it almost impossible to make updates, replace components, or whole systems, without endangering normal operations due to malfunctions, installation and calibration errors, or immature technology.

This research proposes an alternate method to do this technology merger safely. This method uses new concepts, such as the **trust system** and power grid compartmentalization to dramatically increase protection of the network. The great benefit of this method is that it can be implemented gradually. During this thesis, we will transform a SCADA network compartmentalization problem and a trust system strategic placement problem into an optimization problem, by methodically designing a mathematical model and later applying linear programming algorithms and techniques to solve it.

Acknowledgements

I would like to express my sincere appreciation to my thesis advisor, Dr. Kenneth Hopkinson, for his patience, guidance, recommendations, and assistance throughout the course of this thesis effort. Likewise, I would like to thank my committee members, Lt Col Stuart J. Kurkowski, Major Ryan W. Thomas for all their help and advice in putting this thesis together. I am also very thankful to student interns James Haught for providing his expertise in writing the optimizer code and Alex Stirling for helping in many different ways during this research.

Juan M. Carlos Gonzalez

Dedication

I would like to dedicate this work to my children and family for their support and words of encouragement. I also would like to dedicate this research effort to the memory of an aunt who always reminded me of the importance of family during difficult times and that doing what you believe in is right for others is just as important; even when in doing so may reflect poorly on yourself. However, doing this; gives you the peace of mind of knowing you did everything within your power to help. This satisfaction at the end allowed me to be more productive professionally. I miss your early morning calls... Thanks, "Tia"

Juan M. Carlos Gonzalez

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation		Page
SCADA	Supervisory, Control and Data Acquisition System	2
E.O.	Executive Order	8
PCCIP	*Critical Infrastructure Protection*	8
CI	Critical Infrastructure	8
PDD	Presidential Decision Directive	9
DC	direct current	13
GIC	Geomagnetically Induced Current	18
FE	First Energy	21
AEP	American Electric Power	21
MISO	Mid-West Independent System Operator	21
NERC	North American Electric Reliability Council	21
PJM	PJM interconnection LLC	21
RTO	Regional Transmission Organization	21
ECAR	East Central Area Reliability Coordination Agreement	21
MISO	Mid-West Independent System Operator	23
MW	MegaWatt	24
SCADA	Supervisory Control and Data Acquisition	25
COTS	Commercial off-the shelf	27
PLC	Programmable Logic Controllers	29
WAPaC	Wide Area Protection and Control	29
MTU	Master Terminal Unit	31
PC	Personal Computer	31
HMI	Human Machine Interface	31
PLC	Programmable Logic Controller	32
I/O	Input/Output	32

Abbreviation		Page
RTU	Remote Terminal Unit	32
IED	Intelligent Electronic Device	32
LAN	Local Area Network	33
IT	Information Technology	41
IT	Information Technology	44
UW	University of Washington	52
m	meter	57
MILP	Mixed integer linear problem	60
QP	Quadratic problems	60
MIQP	Mixed integer quadratic problems	60
QCQP	Quadratically constrained quadratic problems	60
CNLP	Convex non-linear problems	60
LP	Linear Programming	61
tN	Trust Node	62
CT	Constraint	67
msec	milliseconds	75
secs	Seconds	75

An Efficient and Effective Implementation of the Trust System for Power Grid Compartmentalization

I. Introduction

1.1 Background

One of humanity's greatest mistakes is not to learn from history. It is our duty as a country to take events from the past and used them to better prepare ourselves so that they are not repeated. This is an important principle in general, furthermore it is vital when dealing with our country's critical infrastructure protection. History has shown that we, as a nation, are vulnerable to attacks on our resources and the elements that control them.

The research introduced in this paper affects a sector of our nation that is vital, the critical infrastructure sector. This sector is composed of industries that make our lifestyle, as we know it, possible. Major components include the energy industry, such as electricity, gas, petroleum, in its different phases of the life cycle, water management and transportation management. The list of industries that have been considered a part of nation's critical infrastructure has evolved over the years.

Most of the industries included in the critical infrastructure list have one thing in common; they all require near real-time monitoring. When a malfunction is detected, decisions need to be made immediately in order to prevent major incidents and catastrophes from occurring. An instance of this unique operational characteristic can be found in the electric power industry.

The electric power industry has become so important for our society that it has become the focus of large efforts to prevent power system collapse and electric service

stoppages. However, history has shown that these incidents are not part of a science fiction movie. They are a reality; and when they occur, they have extremely costly consequences. Unfortunately, the demand on the power system increases over time and this increase puts the system at a greater risk of instabilities and collapses. As this thesis will show, there have been instances where minuscule failures have escalated out to larger regions and become major catastrophes. These events unfortunately have not only occurred throughout the world but in our own country as well. These recent collapses have revealed the urgent need to stabilize power systems by incorporating better technologies that enhance not only the local protection at the field level, but also at a system-wide protection level [18].

At the heart of the electric industry and many other critical utility industries, lies a system that is essential to its performance, the Supervisory, Control and Data Acquisition System (SCADA). This system manages these complex networks, often with thousands of nodes monitored. They are capable of reliable and accurate near real-time reactions (sometimes within thousandths of a second) to normal and abnormal variations, and emergency situations as well. Additionally, response time is not the only concern. Accuracy and reliability are key concerns, as well.

The research reported in this thesis is relevant to these two separate but strongly tied systems, critical infrastructures, specifically the electrical utility grid, and the SCADA system.

1.2 Problem Statement

This research will create a method and tool to safely compartmentalized a utility networks, namely SCADA, into regions. The communication within and between regions will be protected by **trust nodes**, which will provide firewall, and intrusion detection capabilities, without disrupting time-sensitive protection and control systems.

The importance of compartmentalization is clearly emphasized by incidents where a proper network topology could have been the difference between a local malfunction and an international blackout. The "*Northeast Blackout of August 2003*" is the benchmark of the possible consequences of a malfunction within a single utility service such as electricity. Even though the initial stages occurred in a small area in Ohio, at the end the blackout affected 10 million people in the Canadian province of Ontario and 40 million people in eight U.S. states. We will see in chapter 2 that the sequence of malfunctions; if they had appropriately isolated the failures in the initial stage the cascading effect could have been stopped at its roots before it became an international incident. Figure 1.1 shows the region where the first malfunctions appeared during the *Northeast Blackout of August 2003*. Perhaps if a different network topology or compartmentalization had been in place, the instability could have been isolated to avoid its propagation to a larger area.

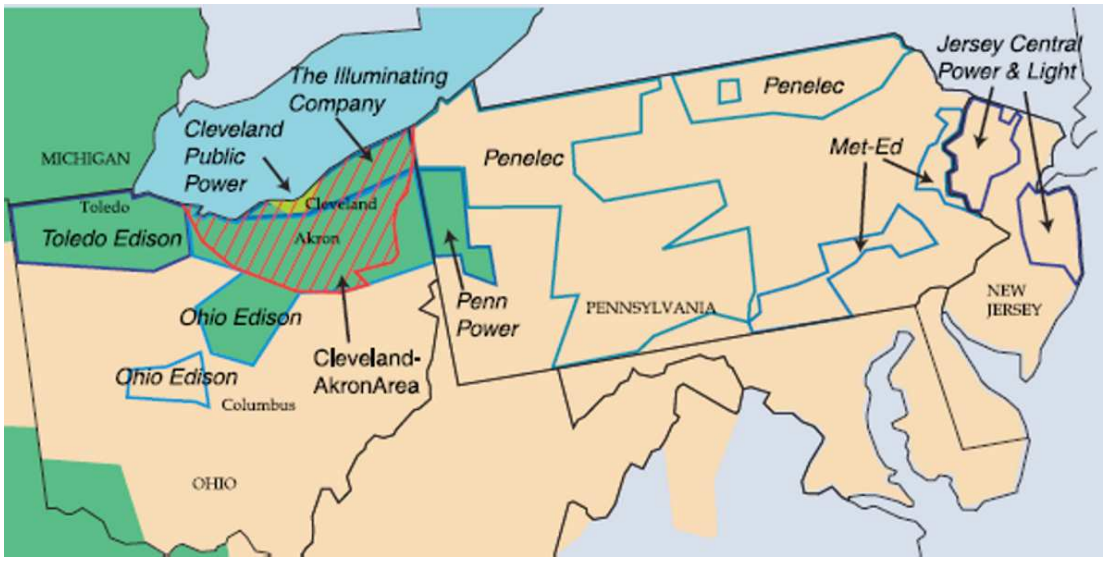


Figure 1.1: Initial Stages Region of Blackout of 2003 [25]

In addition to this, SCADA systems have traditionally been manufactured using proprietary systems and protocols without any regard for interoperability or industry wide standards. However, in recent years the utility community has drifted away from this architecture toward a more open network, communication standards, and

Table 1.1: Sources and Motivations for Utility Disruptions and Attack [9]

Source	Reason
Industrial Sabotage or theft	Financial Advantage in insider trading or competing vendor partnerships
Concentrated physical and cyber attack	Destruction, terror, or activism
Vendor compromise	Easier to target supplier than the defended infrastructure itself
Technical design error or environmental influence	Hardware or code; network design, installation and configuration; or interferences from other technologies in the environment
Natural disasters	Earthquakes, tornadoes, volcanoes, fire, thunderstorms
Operator error	Misjudgment, misconfiguration, or failure to remember operational details resulting in dangerous and costly results

commercial-off-the-shelf components. This has made the industry vulnerable to new threats, such as hacker attacks, other malicious code, and even what is known as cyber-terrorism.

The system is vulnerable not only to traditional equipment malfunctions or operator mistakes. Now, it is also subject to industry deregulation risks and external attacks such as denial of service attacks. Unfortunately, the systems are often configured with little regard for security, and with these new threats security has become a new stronger concern and not just a network burden that increases response times. Table 1.1 summarizes very briefly some of the attacks that the utility industry networks are vulnerable to.

The exposure of SCADA systems to cyber attacks, incidents such as the blackout of 2003 and its cascading effects, and the multiplication of independent companies present in the utility industry, are sufficient reasons to investigate ways to protect our nation's vital functions. As stated in the opening paragraph of this chapter, the approach taken in this research is to investigate the compartmentalization of a SCADA network in an electric utility network. Additionally, we will use the suggestion made by Coates, et al, of a **trust system** that performs important security tasks without exceeding the important time constraints.

1.3 Research Objective and Hypothesis

The purpose of this thesis is to implement and evaluate the suggestions expressed in the thesis document "*Collaborative, Trust-Based Security Mechanisms for a National Utility Intranet*" [5], published by Major Gregory M. Coates, United States Air Force. He suggests the implementation of a **trust system** throughout the power grid and inside a SCADA facility. This thesis will also utilize some of the results from the work "*Evaluating Security and Quality of Service Considerations in Critical Infrastructure Communication Networks*" [28] by Captain Gregory R. Roberts, United States Air Force. He tested the use of several communication protocols at varying background traffic loads. The results incorporated into this research are the refinement of response time thresholds and time constraints.

In addition to combining these two results, the author proposes the compartmentalization (or sub-grouping) of the network (or grid) to provide isolation that may be helpful when a failure is detected and, in that way, minimize or avoid a cascading failure. During the research, **trust nodes** will be added to strategic locations in the networks such that the communication between groups is always monitored and secured. The system will evaluate the propagation delay and also the delay resulting from the installation of the trust node to ensure that no constraint is violated.

It is the hypothesis of this author that the compartmentalized network topologies, and the placement of trust nodes is possible without violating the strict network time constraints necessary for safe operations.

1.4 Why is this research important?

This research is critical, because the systems that SCADA systems monitor cannot be stopped or halted in order to implement security upgrades or install new equipment. The nature of their environment dictates that the system always be operational. This makes their upgrade or replacement very difficult and expensive. The solution evaluated allows for a new security mechanism to be fielded incrementally

without major interruptions of service while increasing the protection the this systems deserve.

1.5 Assumptions

It is assumed that future utility intranet used in power grid and SCADA networks will mimic the network architecture commonly implemented in the corporate world. Some of the delays such as propagation delays, transmission delays, encryption delays are assumed to be accurate for this study based on literature available.

In order to make this research possible several assumptions are considered for the study. I assume the fiber cable is used as the communication means in the grid. This assumption is important in order to calculate the distance between nodes in the grid.

1.6 Preview

The layout of this document is as follows:

- Chapter 2 will provide the basic understanding on the four topic affected by this research, starting from a broader perspective of **critical infrastructure**, then we focus on the **Electric power industry**, **SCADA** network fundamentals, and finally on the concept of a **trust system**.
- Chapter 3 will list the tools utilized during the research and will also describe the methodology and approach taken to solve the problem on network compartmentalization and **trust system** implementation utilizing Linear Programming techniques.
- Chapter 4 will present and interpret the results gathered from the experimentation.
- Chapter 5 presents and explains the conclusions derived from the results obtained during the experimentation and make recommendations for future work.

II. Literature Review

2.1 Chapter Overview

Chapter two has as goals to review the information necessary to understand the concepts examined in this research. It provides the necessary background and complementary material that enables the reader to set the basic foundation , and picture the environment where the results of this research could be utilized. This chapter presents this background information using a top to bottom approach, beginning with critical infrastructure protection, narrowing down to power grid, Supervisory, Control and Data Acquisition (SCADA) systems and finally narrowing down to the specific focus of this research, the description of the trust system in SCADA networks.

2.2 Critical Infrastructure

Infrastructure is defined in the American Heritage Dictionary as "*the facilities, or services for a community to function appropriately*" [20]. These can be the communication system, utility facilities like water and electric power lines, and public institutions to include schools, post offices, and even prisons. Fig.2.1 shows instances of activities that are considered part of our critical infrastructure. In the arena of U.S. public policy, the term has evolved throughout the years, and many times was considered to be ambiguous. In a report issued 20 years ago by the Council of State Planning Agencies, the term was defined as, *"a wide array of public facilities and equipment required to provide social services and support to economic activities"*. In this report, the facilities included roads, bridges, water and sewer systems, airports, ports, and public buildings, and could also include schools, health facilities, jails, recreations facilities, electric power productions, fire safety, waste disposal, and communications services.

In 1984, Congress defined *infrastructure* as facilities with high fixed costs, long economic lives, strong links to economic development, and a tradition of public sector involvement [20]. Hazardous waste services were also added to the list. The con-

(a) Nuclear Plant [25]

(b) Oil Refinery [13]

(c) Electric Grid [14]

(d) Electric Power Generation [14]

Figure 2.1: Examples of Critical Infrastructures

cern was mainly in the adequacy of the infrastructure to satisfy our country's needs. Facilities were many time considered to be obsolete and of insufficient capacity.

Finally, the mid-1990s renewed federal government interest in infrastructure issues due, mainly, to the growing threat of international terrorism. The focus changed from *infrastructure* adequacy to *infrastructure* protection. In 1996, President Clinton signed Executive Order (E.O.) 13010 establishing the President's Commission on Critical Infrastructure Protection (PCCIP) [20]. It was here that the term Critical Infrastructure (CI) was applied for the first time. And it was then that the list of facilities was narrowed down to a few industries which excluded public housing, private rail service, schools, and other facilities. E.O. 13010 ended the ambiguity of the term by listing what it considered to be *critical infrastructure*. According to this executive order, these critical infrastructure were:

- Telecommunications
- Electrical power
- Gas and oil storage and transportation;
- Banking and finance
- Transportation
- Water supply systems
- Emergency services (including medical, fire, police, and rescue) and
- Continuity of government

This list of activities included in the PCCIP final report included facilities owned by private companies and others actually come from other countries such as gas lines that come from Mexico or the electricity distribution lines that come from Canada [15]. This is why our government and more specifically the Department of Homeland Security Science and Technology Directorate conduct research in many areas including cyber security.

As a response to the report, President Clinton signed the Presidential Decision Directive (PDD) 63 [20]. The goal of this directive was to be able to protect our *critical infrastructure* from intentional disruption. PDD-63 directed specific federal agencies to lead this security efforts which are shown in Table 2.1. It is noteworthy to mention the addition of "*cyber-structure*"to the list.

As a result, a national plan for critical infrastructure was created [20]. This plan defines *critical infrastructure* as "*those systems and assets (both physical and cyber) so vital to the Nation that their incapacity or destruction would have a debilitating impact on national security, national economic security, and/or national public health and safety*".

The attacks of September 11, 2001 gave ground to the drafting of E.O. 13228 signed by President Bush and the *USA PATRIOT and Homeland Security Acts* [20]. E.O. 13228 established the office of *Homeland Security and the Homeland Security*

Table 2.1: Critical Infrastructures and Lead Agencies as layed out in PPD-63 [20]

Lead Agency	**Critical Infrastructure**
Dept. of Commerce	Information and communications
Dept. of Treasury	Banking and finance
Environmental Protection Agency	Water Supply
Dept. Of Transportation	Aviation, Highways, Mass Transit, Pipelines, Rail, Waterborne Commerce
Dept. of Justice and or FBI	Emergency law enforcement services
Federal Emergency Management Agency	Emergency Fire Service Continuity of government services
Federal Emergency Management. Agency	Public health services, including prevention, surveillance, laboratory services, and personal health services
Dept. of Energy	Electric power, Oil and gas production and storage

Council. The office is the lead in coordinating efforts to protect our *critical infrastructure* throughout the different government agencies. This Executive Order provides a list of facilities which include previous facilities but it also adds to the list nuclear sites, special events and agriculture to the list which had not been part of the list before. The USA Patriot and Homeland Security Acts are important because they created the *National Strategy for Homeland Security*.

According to the new *National Strategy* there are 11 sectors and 5 key assets to the economy which are considered part of our National Critical Infrastructure, which encompass the following sectors [20]:

- Agriculture and food
- Water
- Public Health
- Emergency services
- Defense industrial base
- Telecommunications
- Energy

- Transportation
- Banking and finance
- Chemicals and hazardous materials
- Postal and shipping

The key assets are the following:

1. National Monuments and icons
2. Nuclear Power Plants
3. Dams
4. Government facilities (offices and governmental departments)
5. Commercial key assets (i.e. major skyscrapers)

As we can see, the identification of our *critical infrastructures* is not an easy task. The term itself has been subject to continuous modifications, economics sectors have been added to the list, while others have been removed. Table 2.2 illustrates how the list has evolved throughout the different directives, executive orders, and other documents that modified due to needs mandated by external situations, such as terrorist attacks and new technologies. The responsibility of identifying these assets has been shared between the private sector and the federal agencies. PDD-63 obligated each federal agency to coordinate efforts with entities in the private sector to assess their very own vulnerabilities to physical and cyber attacks.

Critical Infrastructure is a subject that concerns not only the sectors involved, but almost any other sector of our nation's economic machine. To emphasize its importance, the following lines were added to the official definition, *"An infrastructure so vital that its incapacity or destruction would have a debilitating impact on our defense and national security "* [17].

Critical Infrastructure represents the bird's eye view of the topics covered in this research. The *Power Grid* is the specific system where this research applies.

Table 2.2: Evolution of the Critical Infrastructure List [20]

	U.S. Government Reports and Executive Orders							
Infrastructure	CBO (1983)	NCPWI (1988)	E.O.13010 (1996)	PDD-63 (1998)	E.O.13228 (2001)	NSHS (2002)	NSPP (2003)	HSPD-7 (2003)
Transportation	X	X	X	X	X	X	X	X
Water Supply/Waste Water Treatment	X	X	X	X	X	X	X	X
Education	X							
Public Health	X			X		X	X	
Prisons	X							
Industrial Capacity	X							
Waste Services		X						
Telecommunications			X	X	X	X	X	X
Energy			X	X	X	X	X	X
Banking and Finance			X	X		X	X	X
Emergency Services			X	X		X	X	X
Government Continuity			X	X		X	X	X
Information Systems				X	X	X	X	X
Nuclear Facilities					X			
Special Events					X			
Agriculture/Food supply					X	X	X	X
Defense Industrial Base						X	X	X
Chemical Industry						X	X	X
Postal/Shipping services						X	X	X
Monuments and icons							X	X
Key Industries/Tech Sites							X	
Large Gathering Sites							X	

2.3 Electrical Power System or Power Grid

The electrical transmission system (or Power Grid) developed in North America is one of the greatest engineering achievements of the past 100 years [25]. It connects 200,000 miles of transmission lines which operate at a minimum of 230,000 volts. It has a generating capacity of 950,000 megawatts and it serves over 100 million customer. The infrastructure itself represents an asset worth one trillion dollars.

Originally, power systems were created as self-sufficient units. Power consumption was easily satisfied by the production. In a case of a severe failure, a system collapse was unavoidable and meant a total blackout and interruption of the supply

for all customers. Since the system was small, synchronization of the generators and restoration of the service was easily done [38].

Even though the power system in North America is commonly referred to as "power grid", it is actually divided into three distinct grids or"interconnections". Fig. 2.2 show the interconnections mentioned above. The interconnections are isolated from each other with the exception of small direct current (DC) ties.

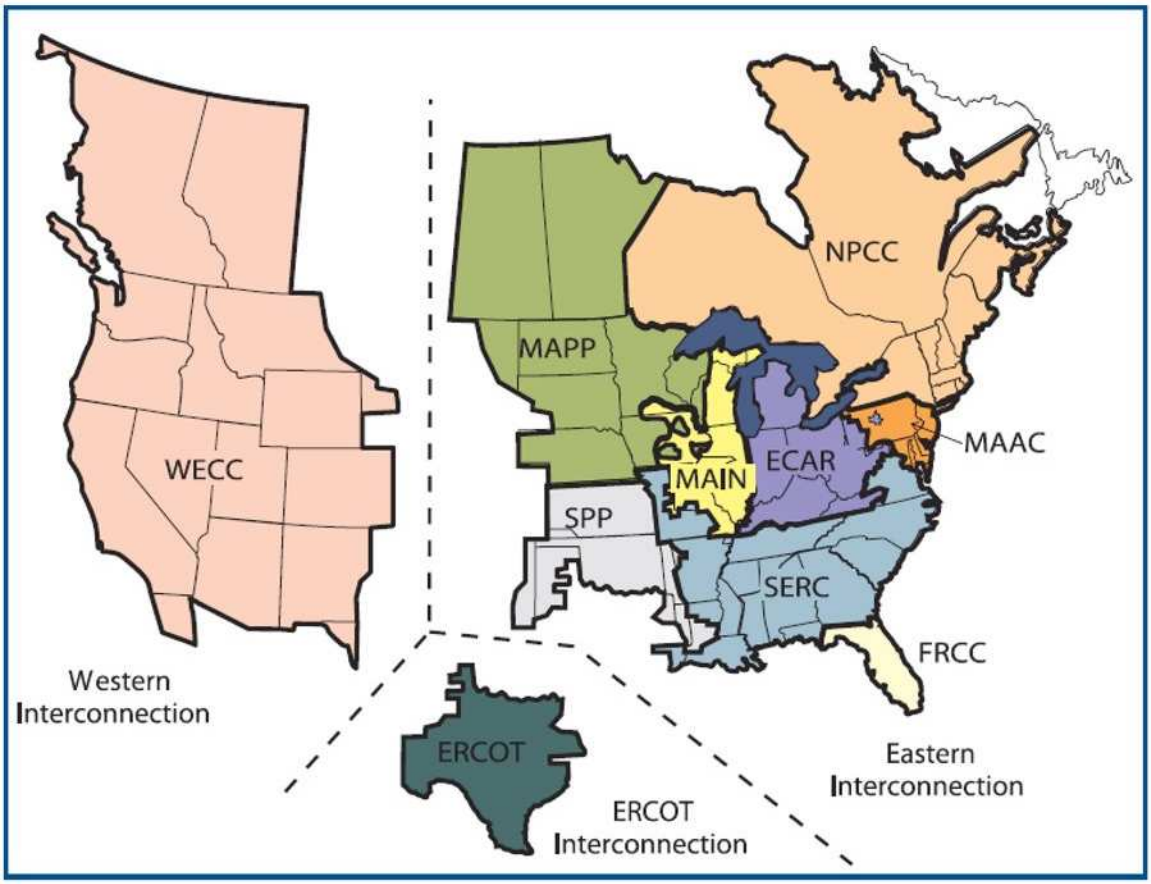


Figure 2.2: North America power grid major interconnections [25]

The reliability of the power grid is crucial to our economy, since our society has come to depend on the electricity it produces for an immense number of activities. Electricity is an essential resource for health, education, transportation, and welfare, as well as national security. Additionally, it powers our heating, cooling, and lighting, computers and electronics, communications, transportations, etc. We have

come to assume the availability of electricity in our daily life and as we integrate new technologies into our daily life we produce more demand for this service. However, we seldom experience blackouts and when we do, they are caused by minor incidents such as a car hitting a power pole, a cable damaged by a lightning storm, or a crew member that affects small areas. But we rarely experience a massive outage covering thousands of square miles and for a period of time larger than a few minutes.

However, reliable electricity presents a complex technical challenge because of the convolutedness of the system of the system that monitors its proper functioning even at normal days and also because of the time constraints that it operates under.

Fig. 2.3 shows the process followed from generation to consumption. At generation stations, electricity is produced at voltages of $10,000$ to $25,000$ volts, regardless of the nature of the generation stations (nuclear, oil, hydro power, geothermal, etc) [32]. Next, it is stepped up to voltages varying from $230,000$ to $765,000$ volts in order to reduce cost and losses when transmitted through large distances. Switching stations and substations provide interconnection between transmission lines. This is named "power grid" because they form a network of lines and stations. Finally, when the energy arrives to the load center it is"stepped down"to lower voltages for distribution to consumers; for industrial and commercial consumers it is normally reduced between $12,000$ to $115,000$ volts and 120 and 240 volts for residential users. All these steps happen almost instantly because the electricity used the moment is generated.

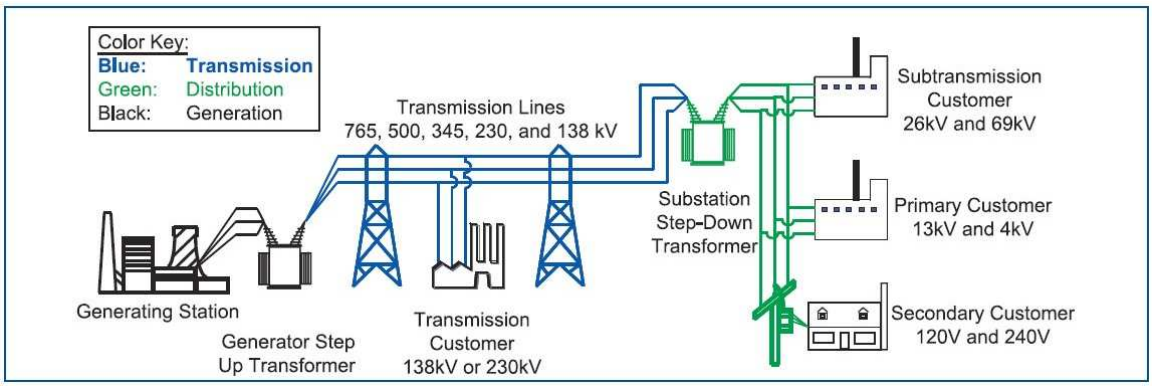


Figure 2.3: Basic Components of the electric power grid infrastructure [32]

As the population grew, consumption grew as well; the demand for power has grown at an accelerated pace beyond that of production. This continuous growth in generation and demand has inherently provoked an increase in the complexity and size of the power system. Unfortunately, these are not the only consequences of this growth. There is also an increased risk of instabilities such as:

- **Frequency Instability**. This situation is caused by the incapability of the system to maintain safe frequency thresholds (i.e. 60Hz).
- **Voltage Instability**. When this situation exists the system is not able to support voltage requirements under normal operations or to recover from disturbances [32].
- **Transient Angular Instability or Generator's Out-of-step Situation**. This condition is present when the system is incapable of sustaining phase synchronization among the generators generally after severe disturbances [32].
- **Local mode of Small-Signal Angular Instability** Similar to the situation described above, however it is present at one station or in a small part of the power grid. Also, it is present under small disturbances which occur continually under small variations in loads and generation [32].

Many times, the system is not able to recover from these conditions, and power outages are sometimes present in small areas and then cascade out to larger regions since the imbalance and instability between demand and generation rises after blackouts occur.

2.3.1 Power Outages. Power outages are also known as power failures, or blackouts. There are several types of power outages, categorized mainly by their duration and the effect of the power loss:

- **Dropout**. This is the shortest of the power losses. Power is restored quickly once the fault is detected, and often the system automatically fixes the fault [32].

- **Brownout**. The name comes from its light dimming effect. This is produced by a drop of voltage. This type of malfunction is particularly damaging to electric motors [32].
- **Blackout**. This is the most severe malfunction, which refer to the total loss of power to an area. Power Outages can last hours and sometimes days, depending on the configurations of the Electrical Grid or the cause of the malfunction [32].

2.3.2 Power Outage Effects. Our society has become completely dependent on Electrical Power, there is no place or activity that does not utilize this source of energy. As a consequence, power failures are particularly critical at sites where the environment and public safety are at risk; such as hospitals, sewerage treatment plants, mines etc [20].

2.3.3 History of Power Outages. Unfortunately, power outages are not rare incidents in modern history. In the last 40 years, there has been a minimum of 20 major blackouts, these include only the wide-scale power outages. The following is a short list of some of the power outages that have occurred since 1965.

2.3.3.1 ***Northeast Blackout of 1965.*** This power outage affected Ontario, in Canada and Connecticut, Mass., New Hampshire, Rhode Island, Vermont, New York, and New Jersey in the United States [35]. The blackout left close to 25 million people and $80,000$ square miles for almost 12 hours. Fig. 2.4 shows the region affected by this incident.

The reason for its failure is attributed to human error [35]. A protective relay on one of the transmission lines was set to a much lower value instead of set to trip and protect the line if the flow of power exceeded the line's capacity. Its origin was located in the Niagara generating station in Southern Ontario.

Figure 2.4: Blackout of 1965 in North America [35]

2.3.3.2 ***Great Storm of 1987.*** On October 15/16 of that year, the most famous weather event of the 20th century in Europe occurred [33]. This storm had gusts between 70 and 100 knots. This became south western England's worst storm since the Great Storm of 1703. It is estimated it killed 18 people in England, 4 people in France and it is estimated that 15 million trees were lost in England alone. The storm left many about $150,000$ households without telephone communications and many hundreds of thousands without power, causing a total of 2.3 million of power disconnection days. Connection days is a measure used by the electricity industry to assess the combination of the number of disconnected properties with the length of the interruption. Fig. 2.5 shows the path that the storm followed.

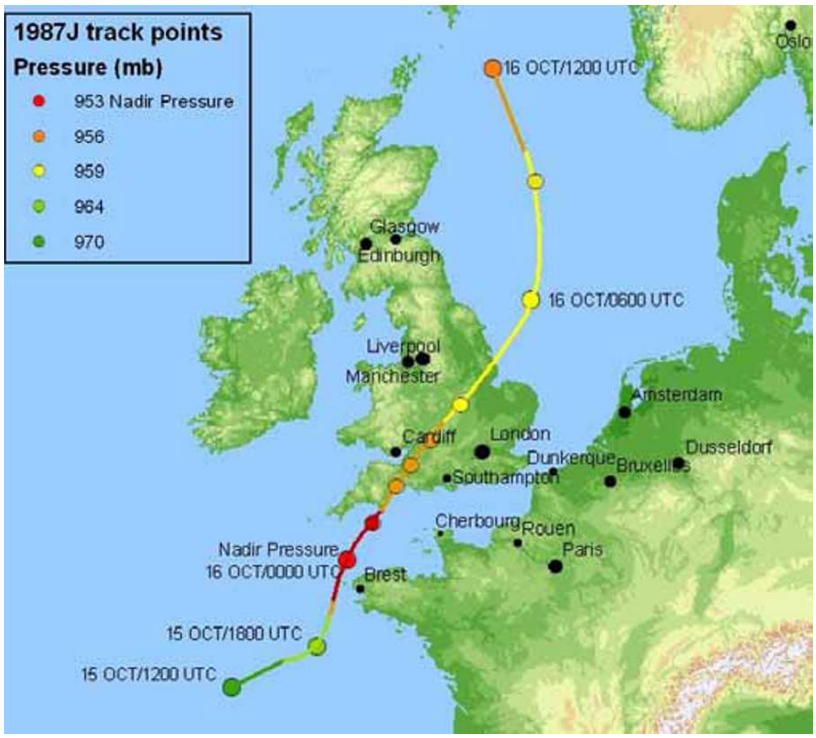


Figure 2.5: Path followed by the Storm on 1989. [33]

2.3.3.3 ***Geomagnetic Storm of 1989.*** A geomagnetic storm is a temporary disturbance of the Earth's magnetosphere which can be caused by changes in the space weather, related most of the time to solar flares and other solar phenomena, which produce a solar wind shock [34] . When the shock travels toward the earth it brings with it pressure changes that disturb the electric field of the earth. The duration of a magnetic storm is normally 24 to 48 hours, however there have been cases where the storm has lasted many days. Fig. 2.6 shows how the sun causes a geomagnetic storm.

Disturbances caused by solar activity can disrupt power grids. When the Earth's magnetic field captures ionized particles carried by the solar wind, geomagnetically induced current (GIC) can flow through the power system, entering and exiting the

many grounding points on a transmission network [34]. GICs are produced when shocks resulting from sudden and severe magnetic storms capture portions of the Earth's surface to fluctuations in the planet's normally stable magnetic field. These variations create potential voltage differences between grounding points, and these cause GICs to flow through electrical transformers, power lines, and grounding points. Unfortunately, only a few amps are needed to disrupt transformer operations. However, over 100 amps have been measured in the grounding connections of transformers in affected areas. It is important to note that many portions of the power grid in North America are vulnerable to geomagnetic storms. Much of the grid is located in northern latitudes, near the north magnetic pole.

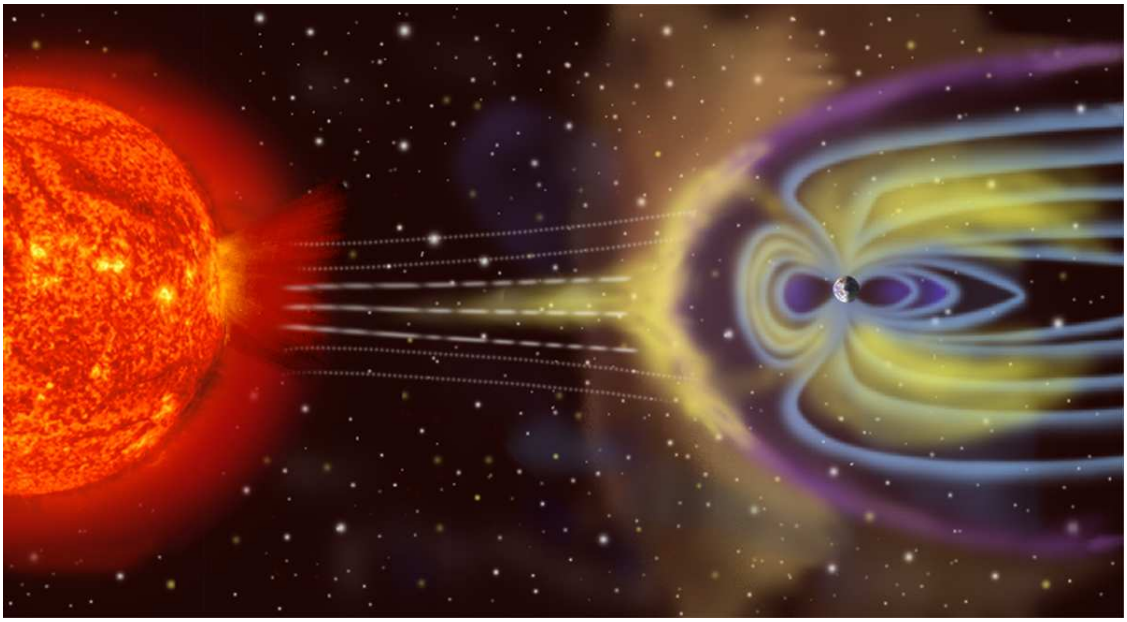
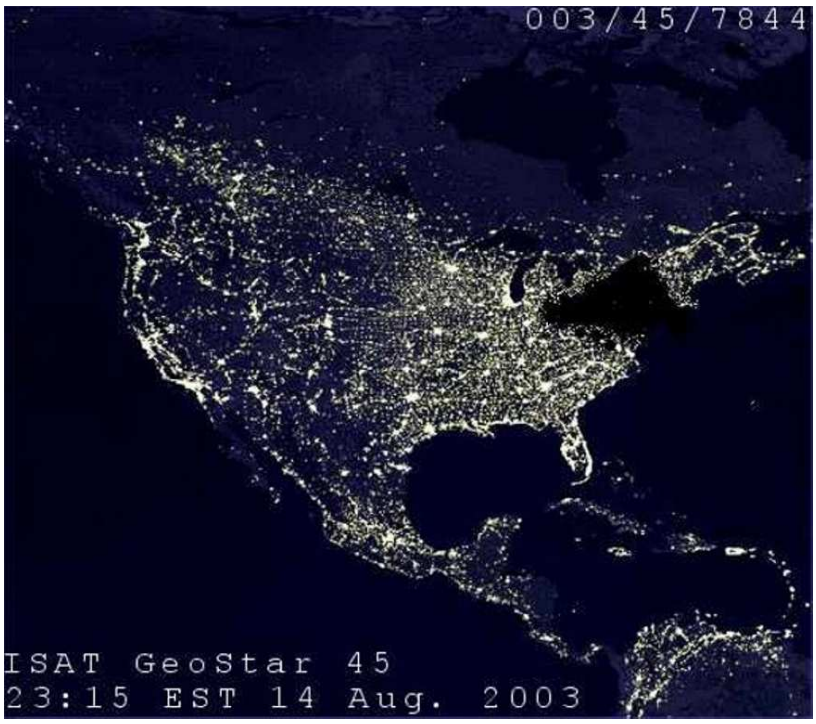


Figure 2.6: Solar Storm and Earth's magnetic Field [34]

The Hydro Quebec outage resulted from the linked malfunction of more than 15 discrete protective-system operations [34]. From the initial event to complete blackout, only one-and-a half minutes elapsed. Fortunately, the outage happened during low demand conditions and was contained within the province's borders. Otherwise, it could have spread across the northeastern United States, extending to Washington, D.C. area.

2.3.3.4 ***Northeast Blackout of 2003.*** On August 14, 2003, large portions of the Midwest and Northeast United States, and Ontario, Canada, experienced an electric power blackout [25]. The outage affected an area with an estimated 50 million people. Power was not restored for 4 days in some parts of the United States. Parts of Ontario suffered intermittent blackouts for more than a week before full power was restored. The estimated loss in the United States range between $4 billion and $10 billion dollars. Fig. 2.7 shows the region that was left without power.



b

Figure 2.7: Region affected by blackout of 2003 [2]

The blackout itself was the consequence of a sequence of events and system weaknesses that maybe by themselves could not have lead blackout of such proportions and could have existed undetected for a large period of time. However, the right conditions existed so that the chain of malfunctions on that day and escalated up to the the collapse of the electrical system.

A group of people from Canada and the United States was directed to lead an investigation that would reveal the causes of the blackout and provide recommendations to stop future incidents like this to ever happen again [25]. The group divided the blackout into phases and identified different causes for each one of the phases but it is important to understand that early malfunctions triggered the later ones. Additionally, the causes were classified based on the nature; some were institutional issues such as deficient practices, lack of adherence to industry policy, and inadequate management; and also, human and equipment failures.

There are several entities that are key during the Black out, each one with different functions. Some of theses entities are:

- First Energy (FE) operates a control area in northern Ohio. This company is composed of seven utility companies in the region.
- American Electric Power (AEP) operates a control area in Ohio just south of FE. This company is both a transmission operator and a control area operator [25].
- Mid-West Independent System Operator (MISO) is the reliability coordinator for a region stretching from Manitoba, Canada in the north to Kentucky in the south, from Montana in the west to western Pennsylvania in the east [25].
- North American Electric Reliability Council (NERC) maintains and develops operating and planning standards to ensure reliability of a transmission grid. This organization is divided into ten NERC regions through out Canada and United States. Fig. 2.8 show the ten regions in NERC and the main connections that monitor the reliability of the electric grid [25].
- PJM interconnection LLC (PJM)is a Regional Transmission Organization (RTO). It is currently the largest wholesale electricity market. Perform reliability coordination functions and along with the MISO are expected to comply with all aspects of NERC Operating policies [25].
- East Central Area Reliability Coordination Agreement (ECAR) is a region within the NERC organization that provides reliability oversight [25].

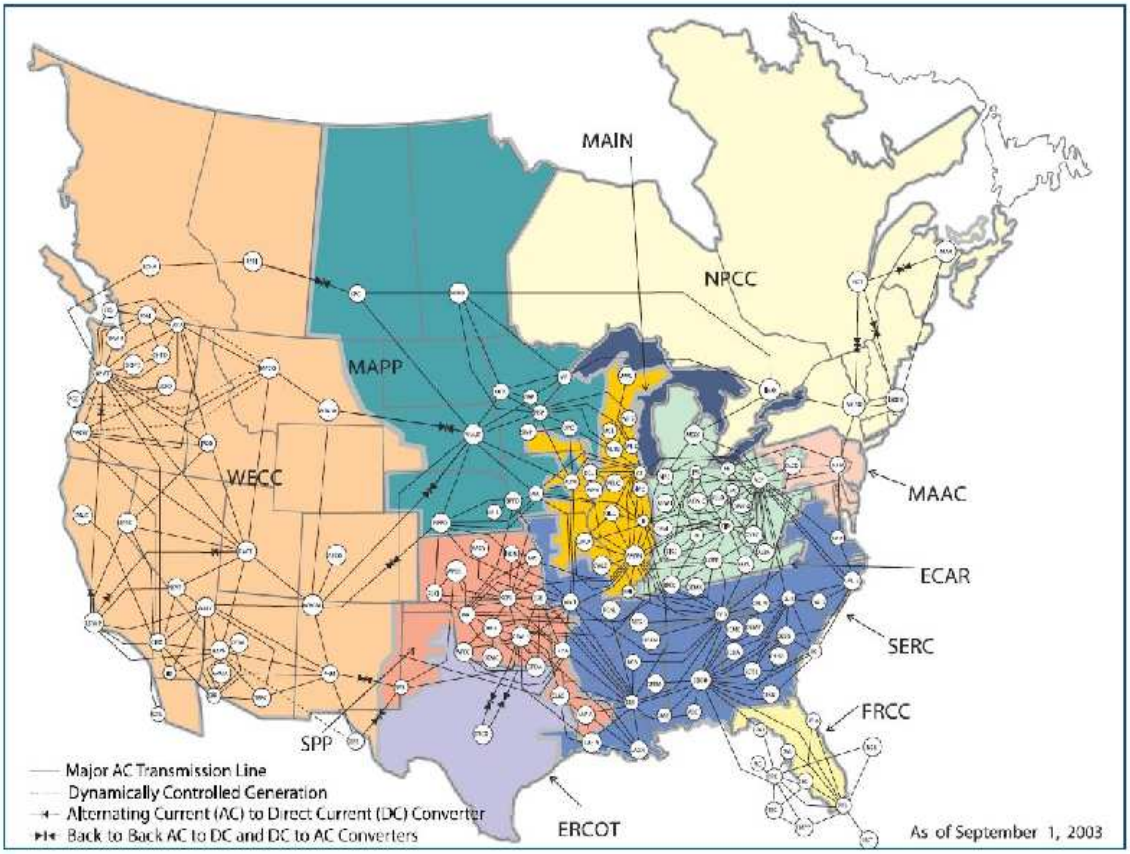


Figure 2.8: NERC Regions and main connections [25]

A large number of causes and/or weaknesses in the sequence of events during the initiation of the blackout have been identified. These have been classified in four groups which are greatly summarized as follows:

Group 1. FE and ECAR failed to assess and understand the inadequacies of FE's system particularly instability and vulnerability of the Cleveland-Akron area [25]. As a consequence, the system operated at inappropriate voltage values. It was found that no review or analysis was done to establish these values and no long-term planning was studies were done in the system. Also, no extreme condition assessments were completely neglected.

Group 2. FE had an inadequate situational awareness and did not understand the deteriorating condition its system. FE did not have the appropriate tools in order

to ensure that monitoring tools were reflecting the functional state of the system, and also lacked back-up monitoring tools to visualize the situation [25].

Group 3. FE failed to ensure that tree growth in its transmission line was properly maintained and monitored, which caused the outage of three FE 345-kV transmission lines and one 138-kV line [25].

Group 4. The Reliability organization at the interconnected grid (MISO) failed to provide real-time diagnostic support [25]. The absence of this real-time data from Dayton Power and Light's Stuart-Atlanta companies prevented MISO from detect security violations in FE's system and execute relief actions. Also MISO lacked an effective way to identify the location and significance of transmission line breaker operations to be aware of important line outages.

The study determined that even though the system was electrically secure minutes before the malfunctions began; there was clear evidence that the Cleveland-Akron areas were highly vulnerable to malfunctions and voltage instability issues [25]. FE was unable to identify the situation because the company had not perform studies to determine and understand those vulnerabilities. FE was operating that system very close to NERC's operational reliability standards. The system stability could have been compromised by any number of potentially disruptive scenarios could have. A system with this little margin to react would leave little room for adjustment, with few relief actions available to operators in the face of single or multiple contingencies.

The following is a quick snapshot of the series of issues that occurred during the *initial phase* of the blackout chronologically, it is not all inclusive:

- It began at 12:15(EDT) when an inaccurate input data rendered Mid-West Independent System Operator (MISO)'s state estimator (a system monitoring tool)ineffective.
- 13 : 31 FE(responsible for the control of Northern Ohio's area), Eastlake 5 generation unit tripped and shut down automatically.

- 14:14, the alarm and logging system in FE's control room failed and was not restored until after the blackout.
- After 15:05, some of FE's 345k-V transmission lines began tripping out because they were touching overgrown trees within the line's right-of-way areas.
- 15:46, FE, MISO, and neighboring utilities begin to realize that the system was in jeopardy. At this point, they could have stopped the cascade effect been avoided by dropping the load around Cleveland and Akron at least 1500 MW. However, no such effort was made [25].
- Moments later, FE lost key lines in northern Ohio which caused its 138-kV line to begin failing and in turn loss of FE's Sammis-Star 345-kV line and it is this event that triggered the uncontrollable cascade portion of the blackout sequence. The Sammis-Star line was critical because it shut down the 345 kV path from eastern to northern Ohio [25].
- By this time, northern Ohio was already blacked out which created an unsustainable burden on lines in adjacent areas. Generating units automatically tripped by protective relay action to avoid physical damage.

  The sequence of events during the initial phase of the blackout is briefly displayed in Fig. 2.9.

There are a large number of other incidents not shown here for brevity reasons, the snapshot shows how much little time it is required for a large area to be left without electricity.

The next section describes the Supervisory, Control and Data Acquisition system. The electric grid system operators must keep close and constant watch on the multitude of things occurring simultaneously on their power system. Because it is not humanly possible to watch and understand all these events and conditions simultaneously, energy management systems use alarms to bring relevant information to operator's attention. The alarms draw on the information collected by the SCADA

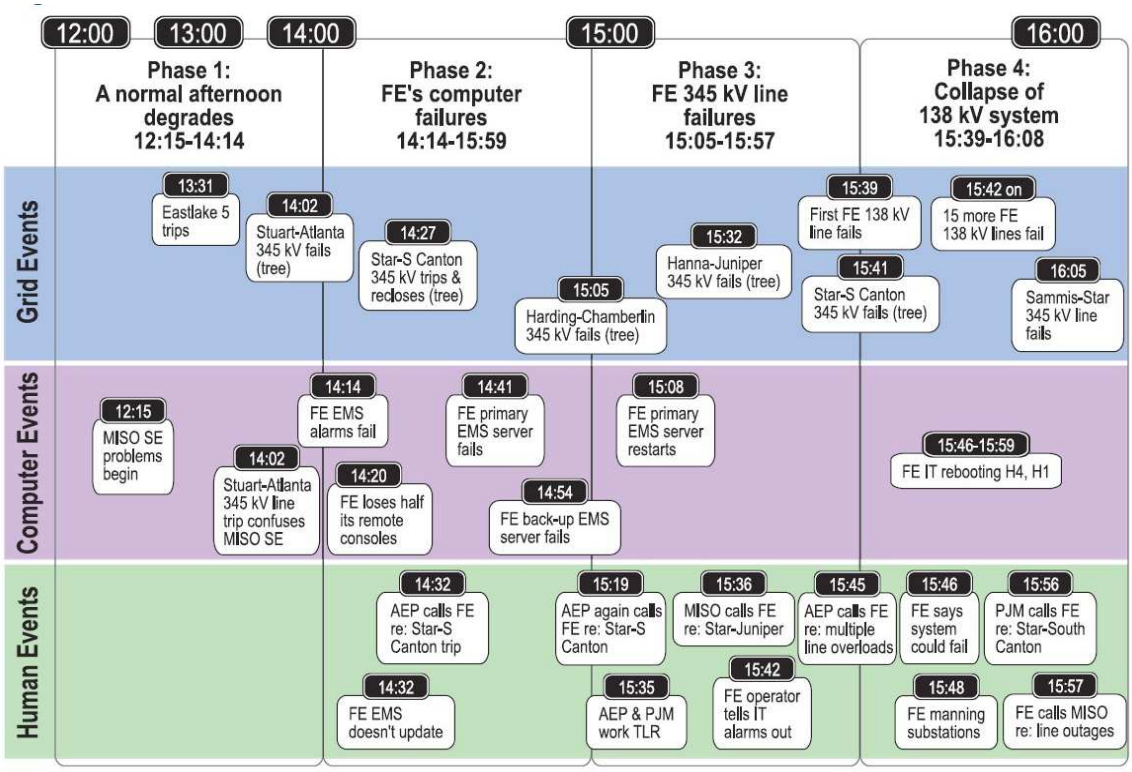


Figure 2.9: Timeline of events during the initial phase of Blackout. [25]

real-time monitoring system. Therefore, SCADA systems are an important component of the electric system.

2.4 Supervisory Control and Data Acquisition Systems

Supervisory Control and Data Acquisition (SCADA) systems are extremely important components to the protection of CI's to include the electric grid. This system is responsible for the safe daily operations of the nation's CI. It controls and supervise systems such as gas pipelines, water a transportation, utilities, refineries, nuclear plants, utilities, chemical plants, and other operations vital to any country's economy. As explained earlier in this chapter, the nature of these infrastructures makes their protection and assurance of availability vital to both the US and world economies.

SCADA allows a central location to control and monitor a spread distributed environment, such as oil, electric, gas field, pipeline system, hydroelectric complex

located hundreds or even thousands of miles from one end to the other. SCADA systems make changes on distant process controllers **(Supervise)**, open or close valves or switches, or monitor alarms **(Control)**; and gather data and telemetry information **(Data Acquisition)** to allow for a secure facility location operation [3].

SCADA provides real-time information to enable the management of production operations, implement more efficient control paradigms, and improves plant and personnel safety. SCADA utilizes communication methods to perform its critical functions such as Directly wired communication, power line carrier, microwaves, radio, and fiber optic communications [6]. This efficient operation of the facility not only provides more economic performance by reducing cost while operating at optimal conditions, but ensure safety of personnel and millions of people around the facility [37]. Fig. 2.10 presents a central control station layout.

The rapid escalation of fuel prices has caused the cost of producing power to escalate rapidly and apparently without control. As a consequence, the efficient and optimum economic operation and planning of utility and, electric power generation systems have always occupied an important position in the utility industry. However, the rising cost is not new to our country; after War World II, the United States began the installation hydroelectric plants to balance the threat of the already increasing price of fuel. Parallel to this new source of energy; thermoelectric and nuclear have also diversified our sources of energy. In addition to this diversification of sources, the introduction of private companies introduced a new variable into this equation. One of the few parameters in the industry that seemed too constant was the utilization of SCADA systems to monitor the efficiency of all these different systems.

The once semi-isolated industrial control offered by SCADA systems which uses proprietary hardware and software are evolving from this standalone, compartmentalized operations into an exposed, networked architectures. SCADA systems have evolved and are now part of a network that has greater control and supervisory capabilities of these facilities. The slowly transforming SCADA system uses standard

Figure 2.10: SCADA master control station [6]

Commercial off-the shelf (COTS) software and hardware. This "standard" system has helped in reduction of development, operational, and maintenance costs. In addition to this, SCADA systems have seen a dramatic improvement in their ability to provide real-time information; critical in the planning, control, supervision, and decision making functions.

The term SCADA actually defines a system that performs the functions described above; however, it is not closely linked to a specific type of hardware, software, or configuration of both. SCADA systems have been developed by a variety of companies that have introduced their own equipment and software. Their version of SCADA may not be inter-operable with other system of the same name. The inter-

national community also referred to the equipment as SCADA however, it does not completely equate to the same system everywhere else.

2.5 Brief History of SCADA

SCADA technology began in the 1960s, when industries began monitoring and controlling instrumentation remotely. SCADA automated systems combine humans, computers, communications, and procedures [30]. They needed to reduce manpower requirements for monitoring of sensors and processes [19]. Early implementations of SCADA systems utilized proprietary software and communications protocols, enormous mainframe computers, and very specialized equipment. Systems not only lacked inter-operable but also difficult to maintain. The industry lacked a standardization mainly due to the high reliability expectations of the system, vendors did not want to rely on someone else's equipment to meet those expectations.

One of the most important function is the SCADA system is *telemetry*. *Telemetry* is the ability to read performance measures from remote locations to evaluate conditions and perform decision making [3]. In addition to this, activities such as weather and geophysical research required the collection of data from places where the presence of a human being was extremely dangerous or not feasible. Or maybe the facility was located in areas where it was difficult to get crews to live, away from populated areas. Although, there was technology available to transport telemetry assets to its remote destinations (i.e., rockets); humans were still needed in the process. This led to the development of communications technologies that allow us to take readings without threatening lives in the process. And it is this communication system that is called telemetry.

In the beginning, telemetry made use of wired communications, sometimes using underground cable. This architecture limited the distance that could be monitored, the number of locations, and also the geographic location where the cable could be buried.

The development of radio signals became the immediate answer to overcoming hardwiring limitations and slowly made his way in the industry [3]. First, it allowed for one way communications allowing only to gather data from to remote site and send to a central location. As a consequence, the central location was not able to send information back to the remote facility. Nonetheless, radio signals have properties that make them very attractive to system developers, such as weather immunity. But the technology was not affordable at the time. As technology evolved and cost reduced more and more, companies incorporated new technology in their daily critical operations.

Along with this, radio signals have been improved to the point where two-way was made possible, this breakthrough allowed the central station to receive data but also to transmit commands back to the remote terminal [3].

As computer technology matured; computers became the heart of the system and a trigger for decentralization of the SCADA structure where it made sense to implement [15]. Control systems consisted of a central minicomputer called Programmable Logic Controllers (PLC) that communicated with local controllers (which could be PLC's as well) that interfaced with motors, pumps, valves, switches, sensors, etc.

2.6 Time Constraints

The response time thresholds under which SCADA systems normally operate are usually very small normally in the range of milliseconds [5]. SCADA is a Wide Area Protection and Control (WAPaC) system, which gathers information from multiple locations on the system and also provides the controls necessary to respond to anomalies detected. The location of those supervisory station is usually at great distances from the anomaly origin location and there may be a time delay. Today's wide area communication structure are capable of delivering messages from one location to multiple locations on the system in as little as 6 ms [5]. This is only one of the different types of delay native to the system. There are several others that are im-

portant to account for such as calculation delay, encryption delay, decryption delay, etc. Table 2.3 shows the customary time constraint thresholds that must be met for SCADA and utility protection responses.

Table 2.3: Typical SCADA Time operating constraints [5]

Systems	Situation	Response Time
Substation IEDs; Primary short circuit protection and control	Routine power equipment signal measurement	Every 2-4 ms
	Local-area disturbance [6]	<4 ms from event detection to sending notification [14]
		4 - 40 ms automatic response time
Backup protection and control; Wide-area protection and control (WAPaC)	Transient voltage instability	Often $\leq$ 180 ms to convey 14+ trip signals to disconnect generators at the top generating station [16]
	Frequency instability, must respond faster than generator governors to trip generators instantaneously	Could require < 300 ms response time (by load shedding) for high rates of frequency decay; requires detection within 100 ms to allow operator response in 150 to 300 ms [16]
	Dynamic instability	A few seconds
	Poorly damped or un-damped oscillations	Several seconds
	Voltage instability	Up to a few minutes
	Thermal overload	Several minutes for severe overloads, rarely less than a few seconds for minor occurrences [16]
SCADA	Emergency event notification	< 6 ms
	Routine transactions	< 540 ms [3]
	Routine HMI status polling from substation field devices	Every 2 secs

2.7 SCADA System Components

SCADA it is not a product of a single vendor therefore configurations are not all exactly the same. Traditionally, each vendor provides its own version of hardware, software, or communication protocol. SCADA systems have been by nature proprietary systems. However, the components have essentially the same function in the system.

Today's SCADA systems are a combination of legacy and modern technology. New technology and components are used along with older ones that have seen small

modification to give room for the new components. The SCADA system can be reduced to a few very major components:

1. **Master or Central station** which houses:

   (a) *Master Terminal Unit (MTU)* also called server or host computer. This is the system controller. The MTU is the center of operation. It monitors the field autonomously, with the proper parameters; it can schedule update requests or perform instructions, and monitor the remote stations based in the current state of the system. The MTU has the capability to monitor hundred of remote locations simultaneously. Depending on the size of the SCADA system, an MTU can range from a single personal computer(PC) to a large room containing dozens of computers and operators [15].

   (b) *Human Machine Interface (HMI)*, presents information graphically to the operator. The operator can normally observe a schematic representation of the plant being controlled [5].

   (c) *Operational databases*, usually linked to the HMI to provide trending, diagnostic data, and management information such as scheduled maintenance procedures, logistic information, detailed schematics for a particular sensor or machine, and expert-system troubleshooting guides [5].

2. **Substations or remote locations**. It is here where most of the supervisory control and data acquisition occurs, and it is done mostly automatically. However, functions are usually restricted to basic site overriding or supervisory level intervention [5].which house:

   (a) *Substation Data Concentrator*, as its name states; it puts together the RTU and IED data from multiple field nodes into a single SCADA address for each SCADA interface with the MTU [5]. The data concentrator polls each IED and/or PLC for updates,then collates the data received from the IED's performs logic calculations, synchronization, data preprocessing so

that this data is sent formatted appropriately for the master control station to translate.

(b) Multiple field devices, such as:

- *Power Equipment*
- *Programmable Logic Controllers PLC*, PLCs scan their Input/Output I/O by reading each I/O point [5]. They are difficult to configure and cannot be used to control other devices or used as master controllers. These are not the best choice when the host field station contains a high number of points to monitor (I/O).
- *Remote Terminal Units (RTU)*. The function of the RTU is to monitor, interpret, execute, and respond to messages received from the MTU [5]. The execution portion of its functionality may in fact be a complex process from sending electrical signals, gather data or actually changing states of equipment in the field. Because of the complexity of its operation the RTUs are based on computer technology. RTU's perform the same function than PLCs or better because RTUs have the intelligence to control processes. The RTU records data, communicate, perform process identification control, and other functions that the PLC is completely incapable of performing by itself. RTUs are capable of controlling process or even multiple process without other devices intervening such as a controller or master RTU. These device has the unique capability of using intelligent logic to execute some of its functions.
- *Intelligent Electronic Devices(IED)*. IED's main function is to process incoming analog signals, convert to a digital form, and resend information via their communication link to a substation automation(SA) controller (also known as a data concentrator) [5]. IEDs can issue control commands to maintain a safe state when irregularities are detected. IEDs are devices that allows one or more processors to receive

and send data/control from/to an external device. Additionally, IEDs can communicate among other IEDs and poll or respond to polls from other IEDs. These are critical components because they provide the integration and automation technology within a substation.

.

3. *Communication Infrastructure* to include modems, radio receiver/transmitters, Local Area Network. (LAN), equipment sensors and actuators [5]. SCADA traditionally depended on internal high speed transmission protocol completely developed for that purpose because of SCADA unique near real-time response required. SCADA transmission protocols are designed to be very compact, and even though protocols are completely proprietary to SCADA vendors; they are standardized among the community. This concept is evolving to move these legacy protocols to operate over standard digital data transmission, such as Ethernet, TCP.

2.7.1 SCADA Data Flow Summary. Data acquisition and monitoring functions begin at the RTU or PLC (and now IEDs) level (Substation); it is at this level that most of the activity. This includes meter readings and equipment status reports. Reports are gathered, pre-processed and transmitted by the data concentrator at the substation and then communicated (Communications infrastructure) to the master control station.

The master control station is comprised of the supervisory servers and software responsible for communication with the field devices in substations [5]. At the MTU the data sent by the data concentrator from field stations is compiled and preprocessed so that the HMI presents the data to the operator in the form of display monitor, controls, and other devices. This way the operator can effectively monitor and if needed, make the appropriate decisions required and interact with the substation field devices. The HMI software runs on client workstations in the control center, this may be a single PC depending on the size of the control station.

2.7.2 A note on Intelligent Electronic Devices. Because of today's advancements in microprocessor technology, a single IED is capable of performing numerous protection, control, and other functions that would require separate RTUs and PLC devices. Therefore, this new piece of equipment is replacing both of these components, which are phased out of the system and replaced by the IED. The IED has increased system reliability dramatically and allowed new system management capabilities such as predictive maintenance, improved planning, and life extensions [5]. Also, IEDs can trip circuit breakers to maintain a steady state when anomalies are sensed. Furthermore, with the use of IEDs local assets are able to poll other local assets or answer poll from other local assets to integrate each separate component in the station and give each component a situational awareness of the stations as a whole. Moreover, IEDs are smaller, require less hardwiring, have more intelligent logic embedded, etc. which make it a better component.

2.8 SCADA applications

SCADA systems are used in a wide number of processes and plants. It ranges from the essential tasks of supervising and controlling the generations of necessary toxic substances, to even optimizing productions lines [15]. SCADA is a main component of our CIs and ensures that those primary components of our economy are running safely on a daily basis. Although this thesis is focused mainly in the SCADA system used in the electric power industry, in essence the principle approached here can be applied in a variety of applications, such as nuclear power generation, and petroleum refining among others. We now briefly describe these two industries in order to enhance the concept of SCADA system [3].

1. Nuclear Power Generation. This Power plant variation is very similar to a conventional electric generation plant (i.e. thermo electrical, fossil fuel, etc), because it generates power with the heat produced by high-pressure steam. As the steam circulates through the system it causes a mechanical energy which rotate generators [23]. The difference between these systems is the fuel used to

elevate the water's temperature to produce steam. Nuclear plants use materials such as Uranium 235 to cause a nuclear fission reaction, in which its atoms are broken down into smaller atoms. This reaction produces a violent reaction and large amounts of energy by means of heat. Water is used to moderate the energy/heat produced by this reaction. In doing this,water is is converted into steam. This steam powers the turbines that generates the electricity. The diagram in Fig. 2.11 shows in a very simple way the process explained above.

A critical characteristic of nuclear plants is that in, contrast to conventional

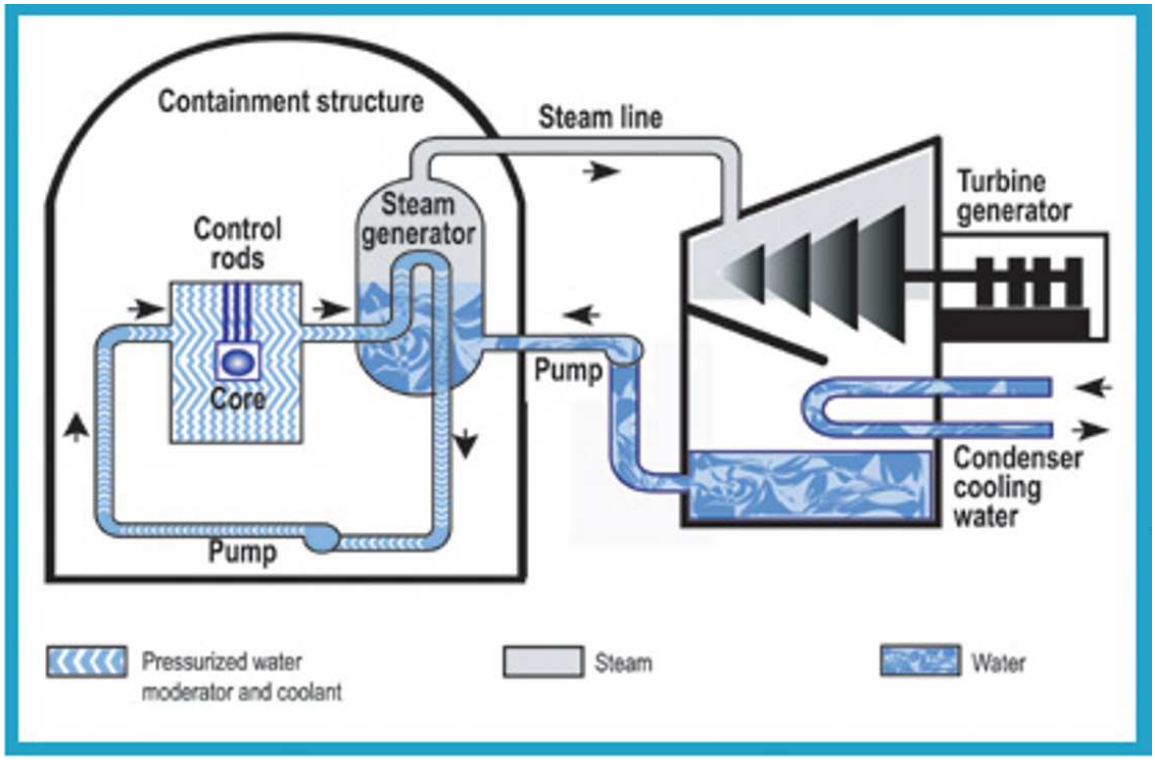
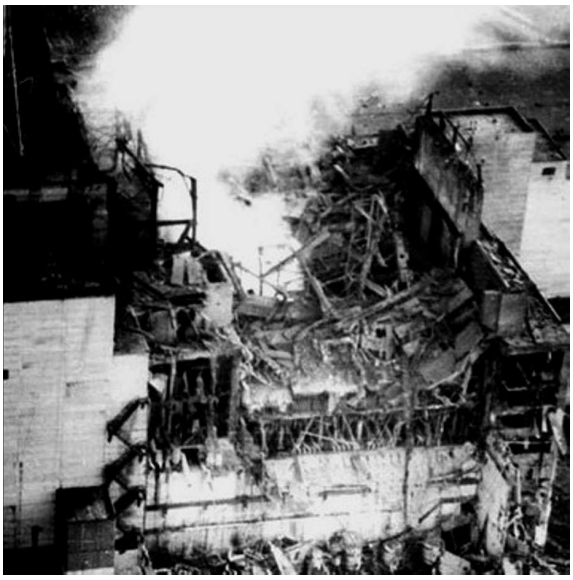


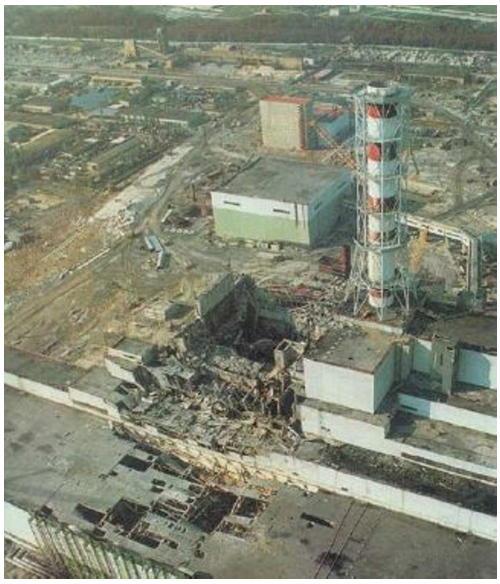
Figure 2.11: Diagram showing how electricity is produced in a Nuclear Plant [23]

sources of energy, a nuclear plant *CANNOT* be completely shut off. Radioactive components are continuously producing a large amount of energy, and their environmental conditions have to be strictly controlled at all times. Water has to be flowing constantly to ensure that the heat produced is removed from the system; otherwise accidents can happen with catastrophic consequences [15].

The Chernobyl disaster is a prime example of why nuclear plants are considered critical infrastructure and how important strict control and supervision is. On April 26th, 1986; the Chernobyl nuclear power plant located in the Soviet Union exploded [12]. The bulk of the casualties were not a result of the explosion but to the radioactive cloud that spread over the atmosphere and carried by winds to remote regions distant from the area. The exact number of casualties and injured is unknown.



(a) Chernobyl Reactor on Fire



(b) Chernobyl Plant after accident

(c) Deformed child due to accident

(d) Monument to Accident workers

Figure 2.12: Images of the Chernobyl Accident [12]

2. Petroleum Refining. Oil is another element of our critical infrastructure. Our nation's economy depends greatly in the supply of this fuel to run almost every aspect of our daily life. Petroleum refineries are extremely important. Services and goods depend on the transportation by trucks, cars, trains, and other vehicles that run on petroleum based fuels [15].

   Refineries satisfy our demand for oil, by operating at high volumes in a constant process. Refineries are designed to handle large capacities and run 24 hour, 7 days-per-week [15].

   The main function of a refinery is to distill and perform other chemical reactions on the crude oil; which require that the system operates at temperatures of 500 to 1,000 degrees Fahrenheit and pressures ranging from 150 pounds per inch to 3,500 [15]. During this process toxic substances in quantities that exceed those tolerable in the finished product, such as ammonia and hydrogen sulfide. These substances require constant monitoring to ensure their removal and safe handling in the process because of their highly corrosive behavior.

   We can see that because of the importance of their final products to our economy, the delicate balance in the distilling process, and the dangerous substances produced in this process; petroleum refineries require constant and strict control during all stages of the operation [15]. An attack in the SCADA system protecting this process could result in fires, explosions, human fatalities, and contamination of large areas. An example of the degree of damage that could be caused by an accident is the explosion of two ships located at a port close to Texas City near a refinery. The ships were transporting ammonium nitrate when they exploded. The explosion caused the refinery to explode and destroy most of the city and killing 576 people. Fig. 2.13 show a a portion of the railroad tracks and the debris that accumulated all around area.

Figure 2.13: Texas City, Texas after the refinery explosion of 1947 [22]

2.9 System Reliability Analysis

As any other system, SCADA has weaknesses, and sometimes these have caused failures that have made evident the importance of the system it controls. The North American Electric Reliability Council (NERC) reported that the system had a total 162 disturbances from 1979 to 1995 [29]. The study analyzed this report and came up with 11 factors that caused these disturbances. Out of the eleven, the three major causes of disturbances or failure are severe weather, unanticipated faults, and equipment failures. The biggest contributors to these disturbances were the real-time monitoring and operating control systems, communications and information systems,

and delayed restoration. All these are highly dependent on a robust information infrastructure and real-time analysis tools [26].

A study done in 2002, point out several bottlenecks in the communications and information system that allowed the disturbances reported [29]. Among them are a lack of automatic communication systems to receive rapid and automatic information, inadequate transmission system security and communication facilities, a lack of advanced communication and emergency communication equipment, and surprisingly, a lack of real-time security analysis and coordinated operation under adverse conditions. This study shows a great need to develop new tools and technologies that enable us to improve the reliability of our utility infrastructure [26].

A strange fact is that SCADA lacks a redundant system to improve its reliability. Geographically separated centers containing backups or duplications could hinder the effects caused by natural disasters or human attacks [15].

2.10 The Threat to Utility Operations

Even though, the NERC reports shows sabotage as a minimum threat, the nature of the industry where SCADA systems performs and along with introduction of the SCADA infrastructure to open/corporate networks raises concern and introduces new threats [16]. Although, there haven't been many documented SCADA system exploits; we can reasonably assume that infrastructure problems through SCADA can occur based on the track record of intrusion to other physical systems.

Technology evolution integration comes at a high cost. The introduction of standard components (hardware and software components) has also presented new vulnerabilities to the system [5]. The same vulnerabilities as a standard corporation network or even a personal computer. Even further, the introduction of the system to the Internet makes it susceptible to intrusions and attacks from hackers from outside as well as internal personnel.

As a consequence, SCADA systems, like any other systems, are becoming more

vulnerable to malicious code such as viruses, Trojan horses, and worms, unauthorized disclosure of critical data, unauthorized modification and manipulation of critical data, and Denial of Service attacks. These threats are critical, we have to remember that SCADA components perform in Real Time or Near Real time, and that their function requires a prompt response to system variations that could cause catastrophic consequences under such attacks.

Additionally, we also have to remember that our country is in the middle of a new type of conflict, asymmetric war; terrorism, where a small number of people with large amount of resources can plan, without being detected, attacks to our infrastructure using any means they see fit. This new threat is in addition to many other threats ranging from conventional direct attack facilities, insider attack (coerced employees), and again Cyber-attacks. The enormous control and supervision area of responsibility of a single SCADA node and the type of facilities under their scope puts them as main possible targets to our country's new enemy, terrorism. Terrorists know that causing system failures to our infrastructure could severely damage not only a single CI but a large geographical region.

2.11 SCADA system security issues

In the report published by NERC in 2002, only 3 out of the 162 SCADA disturbances reported between 1975 and 195 were attributed to security (sabotage). However this could be a misleading result, we have to take into consideration that in the past SCADA was an isolated entity; designed under complete proprietary hardware and software. This type of design is what is named by some as a Closed Design [4]. As it is, SCADA is insecure by nature, because it was never designed with security in mind, other performance factors were given higher priority maybe because the secluded and specialized, equipment, protocols, facilities did not warrant the need to provide a secure system as well [27]. However, the system is now evolving to conventional components; these new trends bring along the vulnerabilities inherent to components not produced for those critical functions.

Control systems are increasingly being incorporated with corporate networks and the Internet. This poses two incompatibilities and security issues by itself. A system as critical as SCADA was not design to be out in the "open" and the Internet does not offer the security required to operate this types of functions.
A study done in 2005 by a SCADA Security Assessment company points out the following list of security issues [27]:

1. Insufficient Network Isolation, loosely defined access controls, SCADA Data integration with IT Systems not secured [27].
2. Insecure remote access; with the new open network, users have access through VPN channels directly of through vendor access, modem connections, or even directly to SCADA end devices such an RTU [27].
3. SCADA flat IP Structure; this type of structure does not protect against malicious codes and does not limit contractor or insider access [27].
4. Vulnerability Risk assessments miss crucial issues; most assessments done are done by firms that ignore the specific requirement of CIs [27].
5. SCADA Security Education/Awareness; Information Technology (IT) professionals do not fully understand CIs and development of a new security certification process directed toward CIs is needed [27].

There are several industrial and government-led (Department of Energy and Sandia Labs) efforts to improve the security of SCADA and control systems in general. A synergy of fields such as chemical, oil, gas, and water, power are concurrently developing programs focused on system security. The electric sector created the NERC which is a standards creation and enforcement body that guides the industry and ensures compliance with these standards [9].

2.12 Utility Industry Intranet

As technology advances are implemented in SCADA systems, new capabilities are added to current SCADA systems. This technology evolution has dramatically

increased the amount of data produced. However, the requirement for real-time communication is still required for the system to accomplish its mission successfully. This particular environment for remote facility management and control lends itself as a perfect candidate for Internet based or Internet like operations. Nonetheless, it is important to note that security and real-time operations are not what the Internet was not designed for or is based on; and unfortunately building an Internet-like system can be extremely expensive [10].

The utility industry has undergone deregulation and the number of utility companies has multiplied as results. Communication between these new players in the industry is important to ensure that operations remain safe. Cooperation between market owners must be paramount to maintaining system stability and reliability [16].

A prime example of this deregulation in the utility industry is the Electric Generation Industry. Nowadays, power companies have been forced to split themselves into different and independent entities with a specific function of generation, distribution, or transmission. The transmission system is typically owned and controlled by the ISO in each region of the power grid. As a consequence, we may find several companies competing for the generation and distribution [10].

This new arrangement poses the problem that since transmission is centrally controlled; only the power grid manager is able to upgrade the transmission infrastructure to meet increasing energy requirements. And this could be a problem, since for some time the power grid has been operating close to its maximum capacity. Also, with this many participants in the industry can make failure detection and isolation in the grid really problematic.

A utility intranet could be designed using many of the standards that the world wide web posses, but placing more emphasis in security. This is understandable because these standards are widespread, low cost, and will ease migration [28]. However, there is already some work being done with this in mind, for example "The Utility Communication Architecture 2.0 and the International Electrotechnical Commission

(IEC) 61850 began lying the groundwork and establishing a specific utility intranet for the industry and some of the power substations are already operating on it, on a limited basis [10].

Finally, there is a new technology that is being widely used in SCADA systems without much security scrutiny, wireless sensors networks. There is, however, work done with the goal of standardizing the communication protocols to ensure confidentiality and integrity mechanisms [16].

2.13 What is next in SCADA

The trend is for even more automation because it lowers costs and increases speed and efficiency. Research and technology development is required to fill the technology gaps between the problems of today and the industry solution of tomorrow. The direction of SCADA is toward fully automated, distributed, and self-healing infrastructures [16]. More intelligence and system level security is needed to eliminate the issues associated with optimizing at a local level and man-in-the-middle limitations.

Also, a point of interest nowadays is the introduction of OLE for Process Control, this is a mechanism for interconneting process control applications running on Microsoft platforms. This new element in SCADA provides better security features that the system lacks currently. It facilitates interoperabilty between a mix of heterogeneous devices in a control network, through a set of common interfaces. However, this product is still under investigation and it is not ready for implementation yet; its development is still underway [26].

2.14 SCADA Security Evolves

Recently, the community of utility companies has begun to shift from the proprietary hardware, software, and protocols that once dominated the industry toward

the adoption of open, networked communication standards for control and data acquisition, patterned after the efficiencies and lower cost of technologies in the Internet.

There has been a constant debate in the industry between power engineers, who have a desire to maintain finely honed processes and speed of operation requirements, and the Information Technology (IT) personnel familiar with network security mechanisms who defend delay-tolerant office networks and see them as the most secure measures for protecting systems against threats such as malicious code and online exploits. Power Engineers raise concern that the majority of common IT security mechanisms used in networks, like the Internet, will upset the current delicate balance in SCADA networks. Both parties are at odds with respect to the role, priority, and implementation of security countermeasures. However, nowadays there are efforts in the Utility Industry guided toward the enforcement of security mechanisms within the Power Grid and inside SCADA networks.

2.15 The Trust System Concept

The concept of a trust system is to provide a non-proprietary system, or software agents that plug into an existing network, somewhat transparently, to perform the functions of correlating data and identifying risk levels for corresponding events and status updates that point to negative impacts on utility services. The trust system, at its core, is a software agent performing active security analysis and response. In a network where nodes have sufficient unused hard drive capacity, memory, and processing power, the agent would be loaded directly onto the node and provide an active interface between incoming messages and the nodes code, data, and applications, similar to other software firewalls. It could also be set to monitor outgoing messages [5].

This collaborative trust system is a hybrid solution comprised of the leading IT security mechanisms and standard IP protocols while focusing on the distinct requirements of the SCADA community, such as the need to allow increased cooperation and

information sharing in protection and control systems without disrupting the critical operation of these systems [5].

2.15.1 How the Trust System works. The trust system intercepts status messages or commands from network nodes destined for the master control station or other nodes in the network. For companies with some legacy nodes, this would require protocol gateway plug-ins for the trust system to interpret and analyze packets delivered in different protocols and formats. The node in the network where one of this devices is placed will be called a trust node [5].

The trust node perform functions of data validation, security risk identification, alert initiation and response actions when bad data is identified. Additionally, it assigns data types to each of the good data elements in each message and determines if the recipient is authorized to read all of the data types in the message. If needed, it sanitizes the parts of the message that are not allowed to be passed to the recipient before forwarding it or simply deletes the message altogether. Finally, the data is then viewable and accessible only to those with the appropriate credentials, need to know, and rights to access those data elements [5].

2.15.2 Inter-Company and Inter-Area Protection. Even though the trust system is not utilized in the present SCADA architectures, this new concept preserves the fine time constraints native to SCADA, but also increases the security protection of the system, which is also very important [5].

The trust system can be placed at strategic locations such as connections between adjacent utility companies, outgoing connections from utility companies to master control stations and engineering centers, and between reliability coordinators would provide low-cost networks security to any of the different types of SCADA configurations. This is important because situational updates shared between adjacent utility companies with different SCADA systems will facilitate automatic recognition of changing conditions that might affect their operations such as load changes versus

current power generation levels. This earlier warning will expedite decisions and response actions such as load shedding or adjust generation rates to absorb or make up for the rapid changed in power flows from adjacent companies [5].

This new capability enables neighboring utility companies to update their operational picture and provides them with a wider perspective of power capabilities and emergency situations. Likewise, control areas can increase their perspective and provide area-wide status and emergency notifications to Regional Utility Operations Center, which in turn improve their regional situational awareness [5].

2.15.3 Internal Traffic Protection. When utilized inside a utility company's network, the trust node provides firewall protections between SCADA nodes and any connected office environment. Moreover, it can ensure fast, reliable delivery of important real-time and emergency traffic.

2.16 Related Work

The use of mathematical modeling or linear programming to solve power grid protection problems is novel. However, there exists several examples of analogous approaches applied to similar situation but unrelated fields.

2.16.1 Combining Quality of Service and Topology Control. This research utilizes linear programming to develop a model that simulates the hybrid wireless network environment [8]. This model accounts for network characteristics such as latency, power consumption, probability of the transmission being intercepted, and priority of the link user. The environment has variables, such as the number of users, the type of links that a user can establish, and user priorities. The problem is defined as finding the optimal network topology, by determining the links that should and should not be establish, given the networks characteristics defined above.

2.16.2 Dialable Cryptography for Wireless Networks. This research objective was to develop an adaptive cryptographic protocol [7]. This protocol takes into

account the hardware and bandwidth available to select the optimal cryptographic strength and algorithm. This idea of dynamically changing the security of a system is important in wireless ad hoc and sensor networks where critical resources such as battery life, memory, computational power and bandwidth are not constant. This research used integer programming to find the best encryption algorithm to use.

2.16.3 Network Design Problem Formulation. The problem approached is to find the design that minimizes the total systems cost defined as the sum of the design cost and the routing cost [24]. Design cost occurs when an edge is added to the network. We assume that we have a flexibility of designing a networks and determining its optimal flow or routing.

The modeling assumption considered in this book, is the "uncapacitated network design problem". Where multiple commodities need to be routed on the network. Each commodity has a source and a destination. The problem is formulated as an optimization problem, where the objective function consists on minimizing the cost

2.17 Chapter Summary

SCADA systems are extremely important for our nation and the world. Critical infrastructures such as the power grid depend greatly on the efficient performance of these type of systems to protect their integrity. WAPaC systems such as SCADA are used in almost every kind of industry but most importantly. According to the Newton-Evans Research Company, 75% of the world's gas and oil pipelines of 25 km or more in length are monitored and controlled by SCADA systems. Application is not limited to CI sector processes.

SCADA moves toward automation because it increases effectiveness and reduces cost. However, Internet's technology was developed ignoring security risks and vulnerabilities for the most part. Unfortunately, from a security perspective, SCADA is as vulnerable as a telephone line can be.

The new trend of utilizing COTS products increases the risk because most of these products were designed for small scale use or for applications not as critical, again in fields where security may not be as crucial as it is in the management and control of our vital infrastructure.

SCADA systems are so important that they impact level I infrastructures – water, power, energy, and telecommunications. However, SCADA is also is also employed in other critical infrastructures such as transportation, food, and agriculture.

Scenarios such as massive power blackouts, oil refinery explosions, or waste mixed with drinking water due to SCADA system compromise, failure, or degradation have the potential to inflict significant damage to human life and critical infrastructure at local, regional, or national levels. If synchronized with a physical attack or the aftermath of a natural disaster, cyber attacks on SCADA systems could greatly escalate fatalities in a region already rendered unable to coordinate a timely response or ill-prepared to offer necessary shelter, clean water, and contamination control, perfect methods for inciting terror once again in America [5].

There have been incidents that have not been widely publicized that reflect the interest of terrorists to attack our industry. For examples, In 2001, the U.S. military discovered evidence in Afghanistan that al-Qaida terrorists were researching SCADA systems [21]. All this information proves that there is an impending need to protect these systems by developing new concepts that improve and enforce security tasks but parallel to this ensure the compliance of environment strict time constraints. The trust system is a viable device adds important security functions. Also, when located at strategic locations trust nodes can stop cascading effects from spreading to larger regions and diminish their consequences.

III. Methodology

This chapter has two goals, the first one is to provide a description of the tools used during the research. The second goal is to describe the methodology adopted to obtain accurate and meaningful results that closely approximate real world conditions and operations. In doing so, this chapter explains the model and scenarios utilized to simulate communications that would be present in the implementation of a collaborative control network (trust system). Ultimately, this research proposes power grid/SCADA network configurations which increase the level of security by adding the trust node security mechanism to the network, by compartmentalizing the network into subnetworks (or domains) protected by these trust nodes placed at strategic locations.

This chapter first describes in broad terms what the problem is, and makes an attempt at emphasizing its importance to our country. Then, it will graphically show what is the research objective by showing how the input network looks like and how it would look, after the input is processed in the optimizer.

3.1 What is the problem?

The United States faces a new type of conflict, terrorism. Terrorists intend to achieve their ideological goals by creating fear or terror by deliberately targeting non-combatants, or any structure that weakens our country. These procedures or "tactics" violate international treaties, and therefore are considered to be unlawful violence and acts of war. The use of unconventional methods presents a new type of threat to our country, to which we may not fully ready. A threat the could potentially attack our country within our borders. The enemy can launch low visibility attacks without the need for a large logistical footprint by utilizing day-to-day equipment and material to perpetrate attacks with catastrophic repercussions. These type of aggressions are mainly directed towards innocent civilian population and *critical infrastructures*.

The protection of our electric power grid is of paramount importance. One way to do this is to develop the technology and/or methodology necessary to strengthen

and protect our infrastructure against these types of attacks. When equipment malfunctions, unauthorized activity is detected, the security infrastructure should isolate the anomaly from the rest of the network to stop it from spreading to larger sections of the network. As a consequence, when a local failure occurs, it does not turn into an incident of major proportions, in the case of the power grid, into a regional blackouts.

3.2 Problem description and Research Objective

The problem described above is broadly introduced here; details are covered in subsequent sections of this chapter. The problem is to convert a power grid or SCADA network topology such as the one shown in Fig. 3.1. This figure shows the input configuration to the model being tested. This network has minimal security protection. It consists of a typical network topology containing the connections between buses or substations.

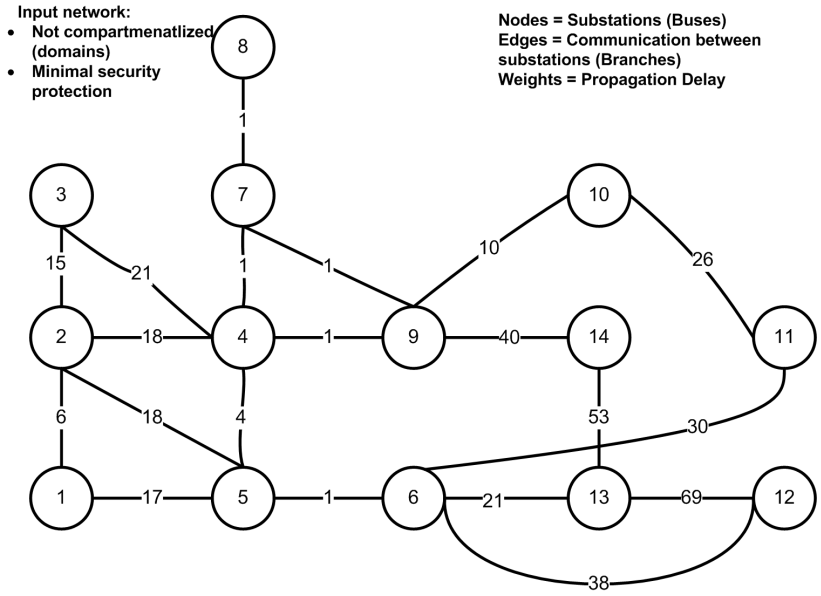


Figure 3.1: Current network topology used as input for this research

The topology shown above is processed and our result will be a configuration showing the characteristics shown in Fig. 3.2. Here, the buses are grouped into

domain and the branches communicating domains are protected by trust nodes (see section 2.15).

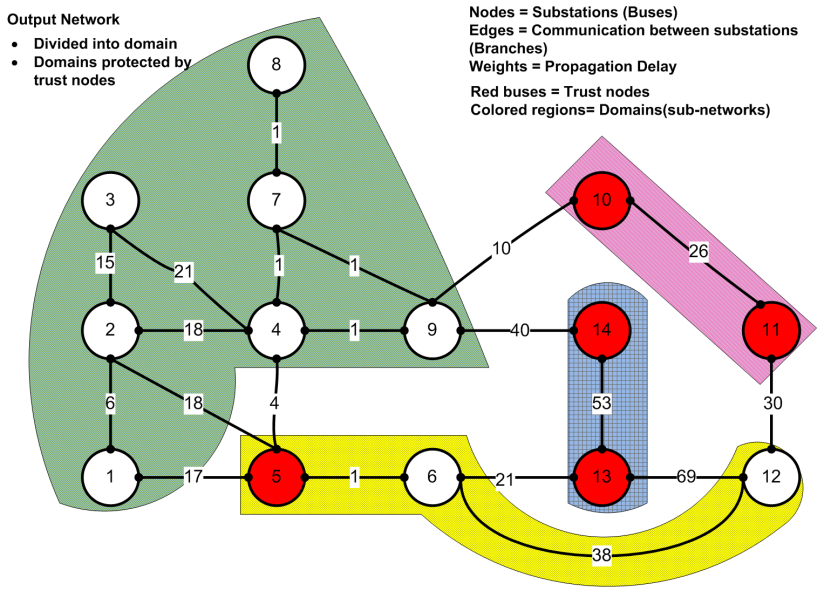


Figure 3.2: Network topology produced showing domains and trust node placement

The goal of this research is to show in a simulated environment that security of the network can be strengthen by first fielding the trust system described above and second, by dividing a network into smaller clusters, called "*domains*," in order to isolate anomalies or intrusions detected. In order to show this, a mathematical model of the problem will be built and translated into a software tool that at the end will receive real-life-network data as input.

This program uses real world power grid representative data, outputs a network configuration that has used the concepts described above of network compartmentalization and strategic placing of trust nodes. For purposes of this research, a node is considered a "*strategic location*" if its positioning within the domain allows the trust node to monitor all traffic (herein called messages) between domains. A solution is feasible if it satisfies the above but also if timing constraint are not violated for any traffic input.

3.3 IEEE Test Case Data

The data that was used for the research was obtained from the University of Washington (UW). The UW Power System Case Archive is a repository of data sets that in some cases represent actual *Power Systems* such as the New England Power System which is represented in the 30 Bus Dynamic Test Case [31]. The data is stored in the standard "IEEE Common Data" format. Each data set has several sections representing information from different devices in the power grid. For example, *branch data*, *bus data*, or *loss zones data*. For the purpose of this research, we will only use the *bus* and *branch* Data. *Buses* represent nodes in the network or substation locations, and *branches* are the connection between buses. There are two different types of data available:

1. Power Flow Systems Test Case Archive. This data set is also called static, and it describes the state of the system at a specific point in time. There are five data sets available with in this test case:

   (a) 14 Bus (nodes). This IEEE Bus Test Case represents a portion of the American Electric Power System in the Midwestern, US. as of February 1962 [31]. Figure 3.3 shows the diagram represented by the test case text file.

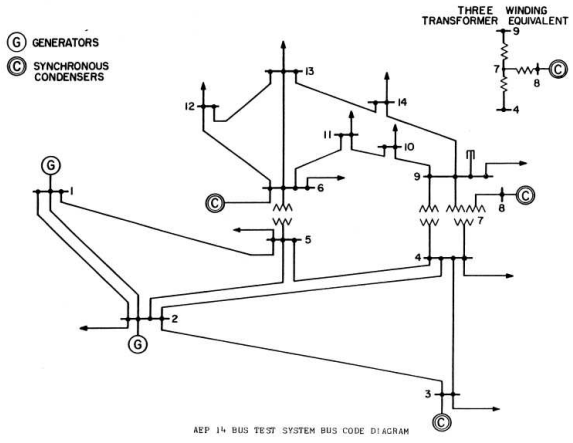


Figure 3.3: Diagram of network represented by the 14 bus test case [31]

(b) 30 Bus. Fig.3.4 presents the network represented by this IEEE Bus Test Case data which represents a portion of th American Electric Power System in the Midwestern,US, as of December, 1961 [31].

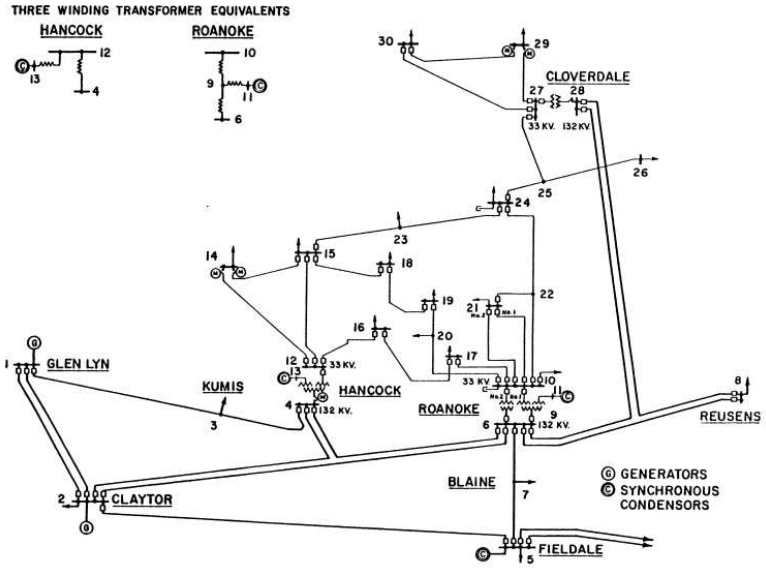


Figure 3.4: Diagram of network represented by the 30 bus test case [31]

(c) 57 Bus. Fig.3.5 shows the IEEE Bus Test Case which represents a portion of the American Electric Power System in the Midwestern,US, as it was in the early 1960's [31].

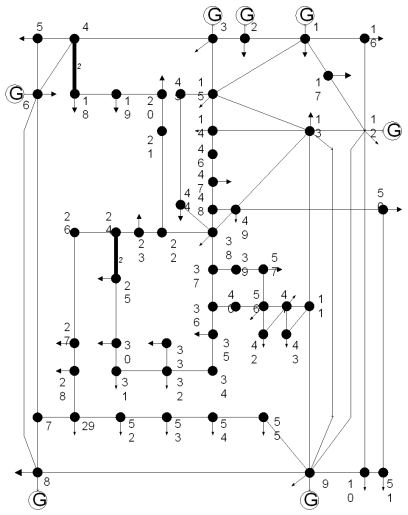


Figure 3.5: Diagram representing the 57 bus test case [31]

(d) 118 Bus. This IEEE Bust Test Case represents a portion of the American Electric Power System in the Midwestern, US, as of December, 1962. Fig.3.6 shows the diagram represented by the test case text file [31].

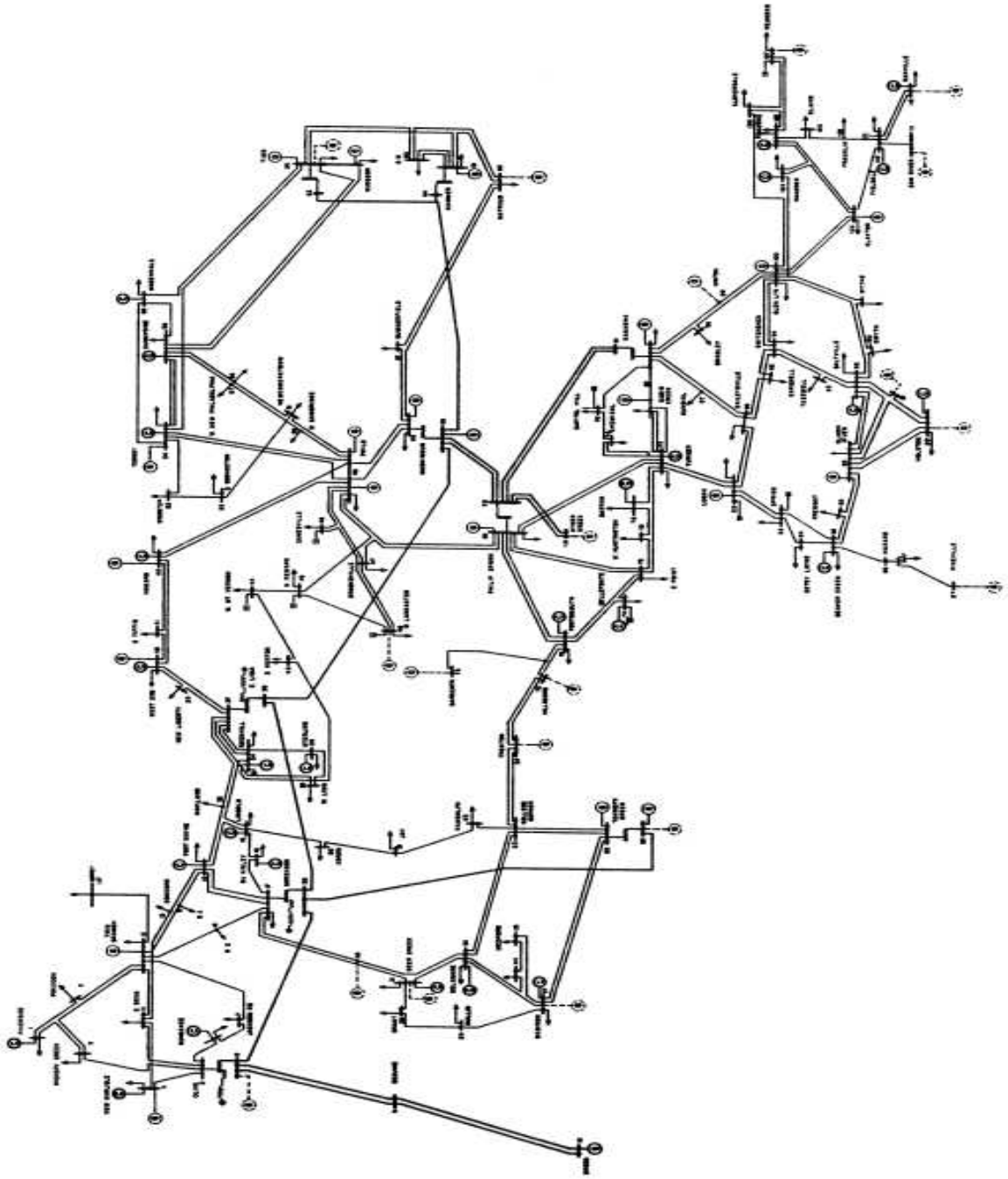


Figure 3.6: Diagram Representing the 118 Bus Test Case [31]

(e) 300 Bus. This data set was developed by the IEEE Test Systems Task Force [31]. However, it is not clear if the data represents an actual power system. Fig. 3.7 presents the topology represented in the test case data.

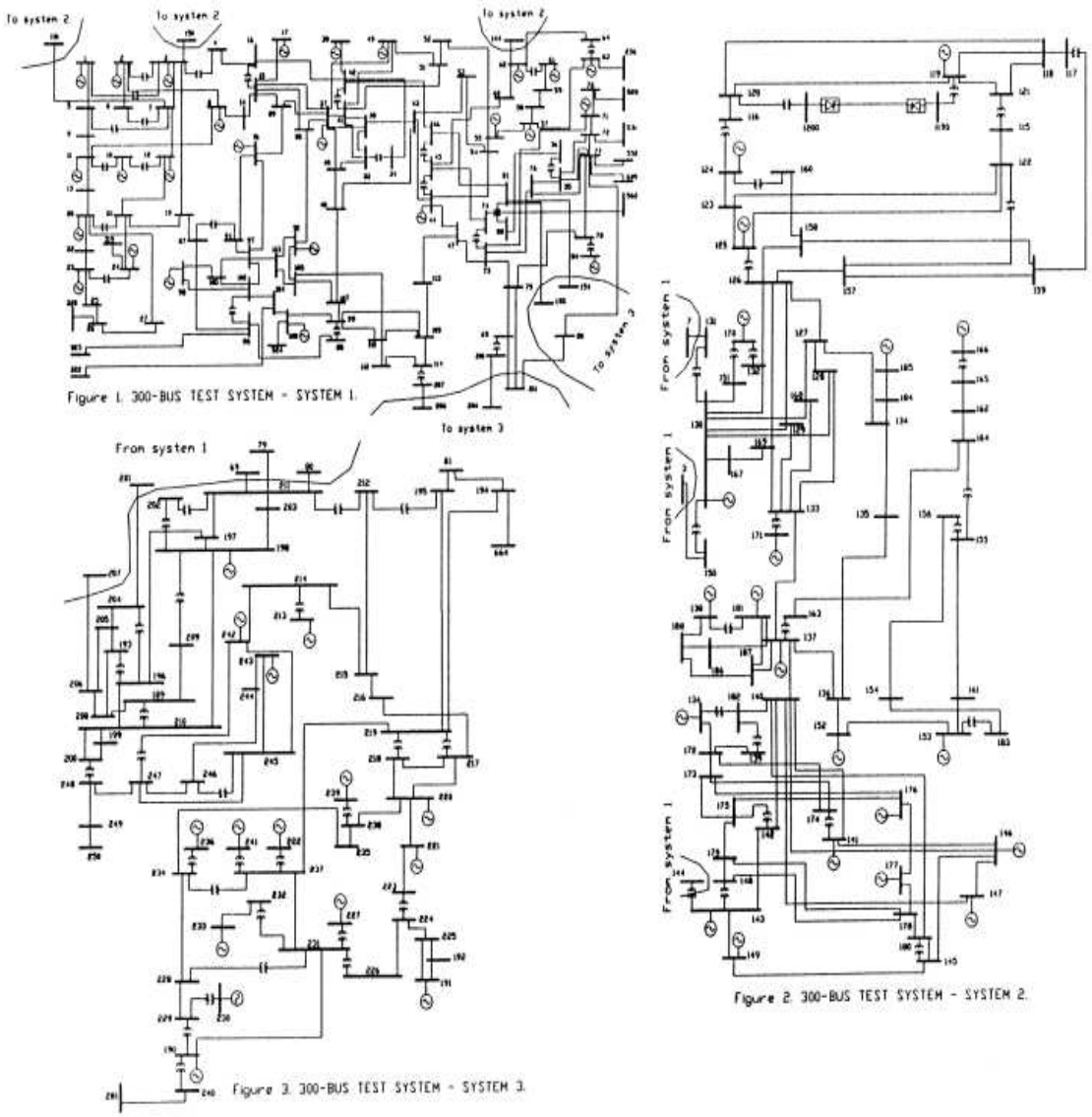


Figure 3.7: Diagram representing the 300 bus test case [31]

2. Dynamic Test Cases. The data set represents the behavior of the system through a period of time. It reflects reactions to voltage variations or other changes that affect the grid and how components reacted to them. This data set was archived from several sources, no diagrams are available for either of these cases:

   (a) 17 Generator, 162 Bus power flow dynamic stability test case, which includes a 162 bus power flow data file. Initially distributed by Iowa State University [31].

   (b) 30 Bus "New England" Dynamic Test Case. This data set was obtained from Arizona State University. And it is representative of New England Physical Power System [31].

   (c) 50 Generator 145 Bus Dynamic Stability Test Case initially distributed by Iowa State University [31].

3.4 Approach

Since our goal is to maximize the number of domains created. Also, the number of trust nodes that can be placed without violating the strict response times(i.e. time thresholds) that the network is bounded by to ensure safe operations. This problem statement fits the description of an optimization problem. In mathematics and computer science, an optimization problem is the problem of finding the best solution (maximum, minimum) within a set of feasible solutions while enforcing system constraints.

Before we begin building a model, we need to make sure that we have all the information/data needed to perform our research. And to do this, preprocessing of the raw test data was needed to make sure it was in the format required to be used as the model input.

3.4.1 Data Preprocessing. The raw data sets do not contain all the information needed to replicate the network with nodes and edges. The data sets contain

what nodes are connected, however, they do not contain the distance between nodes or the delay between nodes. Therefore, additional research needed to be done to derive the distances between nodes (or buses).

In order to derive the distance and/or delay between nodes (stations) a small program was written. The process used in this program to derive distance is described below:

1. From the IEEE raw data, branch section, extract "*Branch Resistance R, per unit*" is stored in column seven of the data. Resistance is measured in ohms, Ω.
2. ρ is the static resistivity (Ω. m). Resistivity is defined below. This value is a constant dependent on the material being used. It is assumed aluminum with iron core a value of $2.50188x10^{-8}\Omega m$. Equation 3.1 shows this formula.
3. *Area* is the cross-sectional area of the material (square meters, m^2). Research show that the Area is 1.25 $inch^2$ which converted to meters is constant value of $.00080642m^2$ was the value used for our calculations [11].
4. l (or distance) is the length of the piece of material (meters, m). This is the information that we are interested on. We obtain distance by using the resistivity formula and solving for l. Equation 3.2 shows the formula used to obtain distance.
5. Multiply this value by 3. The reason for this is that the cable used in the transmission of electricity is composed of three wire [11].
6. Use the absolute value of the number on previous step (distance in meters (m)) to convert to time (delay) in seconds using the formula for velocity, and solve for time. For purposes of this research, we assume that fiber optic is used in the communication line, therefore we used the speed of light $(299,792.458\frac{m}{msec})$. Equation 3.3 shows the formula used to obtain time.

Resistivity (ρ) Also known as *specific electrical resistance* is a measure of how strongly a material opposes the flow of electric current. The electrical resistance of a

wire would be expected to be greater for a longer wire, less for a wire of larger cross sectional area, and would be expected to depend upon the material out of which the wire is made. A low resistivity indicates a material that readily allows the movement of electrical charge. The standard unit of electrical resistivity is the ohm*meter (Ω*m).

The formula for electrical resistivity is:

$$\rho = R * \frac{Area}{l} \tag{3.1}$$

Solve for l and we get:

$$l = R * \frac{Area}{\rho} \tag{3.2}$$

$$Time = \frac{Distance}{Speed} \tag{3.3}$$

In order to better illustrate this, I will describe how a delay was derived.

1. From the raw data set, the program reads column seven from the branch data set portion (i.e. resistance). For this example we will use the actual value $R = .01938\ \Omega$.
2. From our assumptions, we use the resistivity value for aluminum which is $2.50188x10^{-8}\Omega m$ and the cross sectional area of the conducting medium, $0.00080642m^2$. and apply equation 3.2 as follows:

$$l = .01938\Omega * \frac{0.00080642m^2}{2.50188x10^{-8}\Omega m} = 624.69\ m \tag{3.4}$$

3. Multiply by 3 yields $1,874.07m$

4. Apply equation 3.3 assuming the speed of light as follows:

$$Time = \frac{1,874.07m}{299,792,458\frac{m}{sec}} = 6.25124\ x10^{-6}sec \tag{3.5}$$

5. Convert to milliseconds to get:

$$6.25124\ x10^{-3}msec$$

6. Last, we round off to use the integer part of this number (i.e. 6). This number will the used as propagation delay in the model input.

This process was completely automated using the program described above. This program produces the list of nodes and branches connecting them with their corresponding delays. After this it prompts the user to enter desired source node and a destination node, calculate the shortest path and appends it to the network file. With this, we are now capable of taking any IEEE data format test case, derive distances/delays between buses and output the networks input file necessary for our research.

3.4.2 Mathematical Programming or Optimization. In mathematics, the simplest case of *optimization*, or *mathematical programming*, refers to the study of problems in which one seeks to minimize or maximize a real linear function by systematically choosing the values of real or integer variables from within an allowed set or linear constraints [8]. Optimization is a small subset of this field which comprises a large area of applied mathematics and generalizes to the study of means to obtain "best available" values of some objective function given a defined domain where the elaboration is on the types of functions and the conditions and nature of the objects in the problem domain.

In summary; in optimizing a problem the goal is to seek a minimum or maximum value for the objective function by systematically choosing the values of real or integer variables within an allowed set. And this allowed set is defined mainly by the constraints given, which can be set to not to exceed a certain amount or not to go below a set amount of time, money, speed, resources, etc, depending on the nature of the problem.

Mathematical programming has been divided into different subfields depending on the type of degree its objective functions and constraint may have, and the values that the decision variables in the program can take. Some of these subfields are:

- LP - Linear problems have components (i.e. objective function, constraints, and unknown variables) defined as linear functions.
- MILP - Mixed integer linear problem; An optimization problem which involves both integers and continuous variables [36].
- QP - Quadratic problems; It is the problem of optimizing a quadratic objective function of several variables subject to linear constraints on these variables [36].
- MIQP - Mixed integer quadratic problems [36].
- QCQP - Quadratically constrained quadratic problems [36]
- CNLP - Convex non-linear problems [36]

In essence, optimization problems regardless of its subfield are made up of three main components:

1. **Objective Function**, this is the mathematical function that we need to optimize (maximize or minimize), such that we find the "best solution". For example, we may want to maximize profit or minimize cost of operations.
2. **Set of Variables**, which affect the value of the objective function. In a transportation problem the variables can be cost of fuel, distance traveled, number of vehicles available, etc.

3. **Set of Constraints**, which establish boundaries or limits to those variables. For example, monthly budget for fuel expenses. The purpose of these is to define the set of feasible solutions or solutions that fit within the boundaries of the constraints.

3.4.3 Linear Programming. Historically, the first optimization technique is known as *steepest descent* or *gradient descent* which is used to find a local minimum of a function, and goes back to the German mathematician and scientist Johann Carl Friedrich Gauss who contributed greatly to the field of mathematics and several other fields of science [36]. *Linear Programming (LP)* is a technique for *optimization* of a **linear objective function**. This function is bounded by **linear equalities and inequalities** called **constraints**.

LP was developed by *George Dantzig* in the 1940's [36], the term is not in any manner linked to *computer programming*; it was labeled as such because of an acquisition "*program*" by the Unites States military which refers to proposed training and logistic schedules. It was this term that help the project receive federal government funding, since it was immediately associated with high-technology research areas which were considered to be of extreme importance.

When solving problems utilizing this technique a model is created by extracting the characteristics of the problem under the problem domain. Some of the characteristics can be speed, traffic load, number of nodes, distance between nodes, network delay, etc. The technique evaluates the requirements against constraints utilizing linear equations. The goal is to accurately design an objective function that is optimized subject to the constraints natural to the problem being optimized (i.e. response times, propagation delays, etc). The end result should be a set of mathematical expressions collectively called a *mathematical model* that represents the real world problem being solved.

3.4.4 Mathematical Model. The following is a description of this research. First, is the *problem domain* narrative. Most of the information in this section has been discussed in Chapter 2, however the problem domain narrative has not been edited to present the model in its entirety.

3.4.4.1 ***Problem Description.*** The problem consists of an input network that represents a Power Grid or a SCADA facility; the facilities connection and their connection delays. Different types of background traffic are transmitted through the network with corresponding different response times (or thresholds) depending on the criticality of the message traffic. The nodes can represent terminals in the Power Grid or SCADA network components connected by edges, which represent the distance and/or delay between the network components. For our purposes if the network has an edge, (j, k); then a transmission between nodes j and k is possible in both directions. The branches (or edges) between the nodes represent delays that are calculated outside the model. It is important that the delay is always less than the response time expressed as a threshold.

In order to increase the security of the network (i.e. grid, or SCADA facility) the layout needs to be subdivided into *Domains*. This compartmentalization isolates attacks or malfunctions so that they can be dealt within the domain affected. This prevents a rapid cascade effect through out the grid/facility.

Additionally, a new security mechanism, called "trust node" (tN) is "installed" at strategic buses. This device inherently adds delay to the network due to its functioning. The number of trust nodes is limited. As clarification; even though it is called a trust node, it *IS NOT* a node or bus, it is a device that is installed at the bus to add security to the communications flowing through that bus in the network. tN's represent pieces of hardware/software that is added to that node, to increase security functions. Section 2.15 has a more detailed description of the trust node or trust system.

The goal is to be able to partition the network in as many domains as possible (*maximize*), and install as many trust nodes as possible without causing any type of traffic to exceed its corresponding response time. The response time is critical for a safe operation of the facility.

The input is a text file that provides the components of a graph such as a set of branches (or edges), a set of buses (nodes), a set of delays in those branches, traffic represented by its path and its type. Also, the input provides constants that are used in the optimization process.

Algorithm domain description:

Consider a Network represented by a graph G, and nodes

- Let *Graph: G(Bus, Branch)* Where:
  - Let *Bus* denote the set of Buses (i.e. Vertices).
  - Let *Branch* denote set of Branches (i.e. Edges). This variable is defined as an array of integers with a capacity determined by *bus*.

The specific variables utilized in the model are the following:

- Input Variables:
  - Let *nBus* be the integer number of buses in the network.
  - Let *nBranch* denote the integer number of branches the network has in it.
  - Let *nPath* denote the integer number of paths in the network.
  - Let *nTrafficType* denote the different type of traffic types in a SCADA system
  - Let *responseTimes* denote the response times the need to be enforced for each type of traffic in a SCADA system. These are obtained from previous.
  - Let *tDelay* denote the delay added to the network when a trust node is added to a bus.

- The indexing sets are:
  - Let *bus* denote the indexing set for buses. This is initialized as an unbounded range.
  - Let *busRange* denote the indexing set for buses.
  - Let *maxDomainNum* denote the maximum number of domains that can be created.
  - Let *DomainRange* denote the indexing set for the number of domains.
  - Let *minNode_InDom* denote the minimum number of buses (or nodes) that are needed to be assigned to a domain.
  - Let *max_TNode* denote the maximum number of trust nodes available.
  - Let *tNodeRange* denote the indexing set for the set of trust nodes.
  - Let *pathRange* denote the indexing set for the set of paths.
- Let *delay* indexed by the set of buses represent the delay in the branch connecting those buses.
- Let *busInDomain* denote the domain number where each bus has been assigned to.
- Let *domBranch* denote the incidence matrix for each domain created, reflecting only the buses and branches included in that specific domain. This is represented with a three dimensional variable; where the third dimension corresponds to the domain number. The first and second dimensions represent the buses in that domain. This matrix is assumed to be bidirectional.
- Let *domainNodeCnt* denote the count of the number of buses/nodes contained in a domain. This variable is initiated as follows:

$$\forall k \in domainRange : domainNodeCnt_k = \sum_{j \in busRange} busInDomain_{j,k}$$

- Let *totDomain* denotes the count of all domains created.

- Let *unusedBranchs* denote the branches that are not inside or part of a domain. These could be used to calculate domain entry/exit points or buses where a trust node may be placed.
- Let *tNodeLocation* denote the node number where a trust nodes device has been installed.
- Let *tNodeLocationTot* denote the sum of trust nodes used.
- Let *tNodeCnt* denote a count of trust nodes being placed or utilized.
- Let *trustDelayArray* denote the added delay to each path by the total sum of trust nodes in that path. It is initialized by multiplying the delay incurred by adding a trust node and the *tNodeLocationTot*, such as:

$$\forall p \in pathRange,\ \forall j \in busRange :$$
$$trustDelayArray_{j,p} = tNodeLocationTot_j * tDelay$$

- Let *trustDelayResult* denote the sum of the delays incurred when accounting for the trust nodes installed along the path of the traffic. This variable is Initialized as:

$$\forall p \in pathRange : trustDelayResult_p = \sum_{k \in busRange} trafficTrust_{k,p} * tDelay$$

- Let *traffic* denote a matrix containing the traffic path incidence matrix, or in other words, contains the edges included in a path.
- Let *trafficType* denote a numeric array whose values determine the traffic type for the path in the same index position in the variable *traffic*. This number is looked up in the *responseTimes* variable to evaluate the appropriate threshold. For Example, if *trafficType* contains a 3 in location 1, then the path in location 1 of the variable *traffic* is of type 3. And it will use the threshold stored in the variable *responseTimes*(3).
- Let *trafficIncidenceArray* denotes the nodes included in the path being evaluated, the values in the array are 0 or 1 (Boolean Matrix).

$$\forall p \in pathRange : trafficIncidenceArray_{j,p} = \sum_{b \in busRange} traffic_{b,j,p}$$

- Let *trafficDelayIncidence* denote the incidence matrix of the traffic delay incidence after path input. This variable stores the delays incurred during a path. It is initialized by multiplying the delay between two nodes j and k times the positions in the array traffic where a 1 has been stored. This initialization is accomplished by:

  $\forall j, k \in busRange,\ \forall p \in pathRange :$

  $$trafficDelayIncidence_{j,k,p} = traffic_{j,k,p} * delay_{j,k}$$

- Let *trafficDelayIncidenceArray* denote the sum of each of the delays per path. It is initialized utilizing the delays on each node as follows:

  $\forall p \in pathRange,\ \forall j \in busRange :$

  $$trafficDelayIncidenceArray_{j,p} = \sum_{k \in busRange} trafficDelayIncidence_{k,j,p}$$

- Let *trafficTrust* denote the delay added to the path when it traverses a node that has a **trust node** in it.

- Let *trafficDelayResult* denote the total delay per path. This is calculated by adding the sum of the delays on the branches included in a path or traffic. This is initialized as follows:

  $$\forall p \in pathRange : \sum_{k \in busRange} trafficDelayIncidenceArray_{k,p}$$

- Let *trafficResult* denote the delay on a path. This is calculated by adding the delay incurred by the traffic alone and the delay incurred when a trust node was added to the path. Since this delay stores the total delay, it is constrained to be less than threshold. It is an array and each element represents each path's total delay.

  $$\forall p \in pathRange :\ trafficResult_p = trustDelayResult_p + trafficDelayResult_p$$

The following assumptions are made:

- Delay is bidirectional:
  $delay_{i,j} = delay_{j,i}$
- Branches are bidirectional:
  $\forall m, n \in bus : n \leq m \ \Rightarrow branch_{m,n} = branch_{n,m}$
- The incidence matrix for each domain:
  $\forall d \in domainRange;\ n, m \in busRange : domBranch_{m,n,d} = domBranch_{n,m,d}$

  Subject to the following constraints(CT):

- – The Total delay (trafficResult) on every message (traffic) has to be less than or equal to the response time allowed by the system (threshold). The response time is different depending on the type of traffic. (CT 1)
  – The number of nodes per domain can either be 0 (if not being used) or greater than or equal to *minNodesPDom*(i.e. semi-continuous). (CT 2)
  – Every bus has to be assigned to exactly ONE Domain(Sub-Network) (CT 3)
  – Every node has to be assigned to a domain.
  – The number of unused branches has to be greater than the difference from the total branches in network and the sum of the branches that have been assigned to a domain.(CT 4)
  – The number of buses assigned to domains has to be less than or equal to the total number of buses. (CT 5)
  – Every trust node may only be assigned to one bus in the network.(CT 6)
  – The sum of the buses in each domain is less than the total number of buses in the network
  – The sum of all buses on each domain is equal to the total number of buses in the input Network (Bus).(CT 10)

- Every node in the network can have a maximum of ONE trust node assigned to it. (8)
- Every bus can only contain at most ONE trust node. (CT 9.iv)
- *busInDomain, domBranch, tNodeLocation, tNodeCnt, totDomain, unusedBranchs* are binary variables. (CT 9)

Objective Function: Maximize the number of total domains created.

3.4.4.2 ***Mathematical Model:***

Objective Function:

$$Maximize \sum_{d=1}^{domainSize} totDomain_d$$

Subject to:

1. $\forall p \in pathRange: \ trafficResult_p \leq threshold$
2. $\forall k \in domainRange : (domainNodeCnt_k > minNode\_InDom) \vee (domainNodeCnt_k = 0)$
3. $\forall j \in busRange : \sum_{k \in domainRange} busInDomain_{j,k} = 1$
4. $\forall j,k \in busRange : \ unusedBranchs_{j,k} \geq branch_{j,k} - \sum_{d \in domainRange} domBranch_{j,k,d}$
5. $\forall k \in domainRange : \sum_{j \in busRange} busInDomain_{j,k} \leq nBus$
6. $\forall j \in tNodeRange : \ tNodeCnt_j \leq 1$
7. $\forall b \in busRange : \sum_{t \in tNodeRange} tNodeLocation_{t,b} * \frac{nBus}{2} \geq \sum_{k \in busRange} unusedBranchs_{k,b}$
8. $\forall b \in busRange : \sum_{t \in tNodeRange} tNodeLocation_{t,b} \leq 1$
9. Binary Variables:
   i. $\forall j,k \in busRange : unusedBranchs_{j,k} \in \{0,1\}$

ii. $\forall j \in tNodeRange : tnodeCnt_j = \sum_{k \in busRange} tNodeLocation_{j,k} \wedge$
$tNodeCnt_j \in \{0,1\}$

iii. $\forall j \in busRange,\ \forall k \in domainRange : busInDomain_{j,k} \in \{0,1\}$

iv. $\forall j \in tNodeRange,\ \forall k \in busRange : tNodeLocation_{j,k} \in \{0,1\}$

v. $\forall j,l \in busRange,\ \forall k \in domainRange : domBranch_{j,l,k} \in \{0,1\}$

vi. $\forall d \in domainRange :\ totDomain_d \in \{0,1\}$

10. $\sum_{j=1}^{busRange} \sum_{k=1}^{domainRange} busInDomain_{j,k} = Bus$

3.4.5 Application used for model development. The next step in the process was to determine what type software was needed to perform optimization and translate the mathematical model defined above into a language. To accomplish this, the optimizer **Xpress-MP** environment was used. This environment implements a language called **Mosel** [1].

Mosel is a language that is both a modeling and a programming language. This allows the environment to combine the modeling and the programing of the algorithm. Mosel language code is then processed using what is called an optimizer, in this case *Xpress-Optimizer* which is what takes the language and solves the problem represented in it [1].

Mosel allows the user to define models in a form that is close to algebraic notation and to solve them in the same environment. The optimizer utilizes several algorithms to solve problems:

1. Simplex methods: In an LP problem, the region defined by a set of linear constraints is known as the feasible region. The simplex method is based on the fact that the optimal solution lies on the boundary of the feasible region. Usually, simplex methods consider solutions at the vertices on the boundary of the feasible region and proceed from one vertex to another until an optimal [8]. solution has been found, or the problem proves to be unfeasible or unbounded.

(a) Dual simplex methods. The dual simplex algorithm is usually much faster than the primal simplex algorithm if the model is not infeasible or near infeasibility [8].

(b) Primal simplex method, however, is usually the best choice for problems that are likely infeasible as it makes determining the cause of the infeasibility less difficult [8].

The difference between the primal and dual simples methods lies in which vertices they consider and how they iterate.

2. Newton Barrier method. This is an interior method because it iterates moving from one point to the next within the interior of the feasible region. Approaching the boundary of the region is penalized, therefore the process cannot leave the feasible region [8]. Interior point methods usually give a solution lying strictly within the interior of the feasible region, this solution can only be an approximation to the true optimal vertex solution. As a consequence, how close we want to be to the optimal solution and not the number of decision variables, influences the number of iterations required to reach that optimal approximation. This method usually completes in a similar number of iteration as the simplex method, regardless of the problem size.

3.4.6 Input of Model in Optimizer and Validation. As the model was being entered, the approach was to subdivide the problem into different phases and build on the previous phase to implement the new phase. Each one of the phases represent each one of the goals that were needed to accomplish the optimization. The phases that the model was divided onto are the following:

i *Subdivision of input network into domains.* Nodes inside a domain have to be contiguous.

ii *Placement of trust nodes in strategic nodes.* We want the model to place the maximum number of trust nodes in the network. However, trust nodes have to

be located in nodes where they can monitor incoming and outgoing messages between domains. Moreover, the model has to ensure that by placing these trust nodes the response time threshold is not violated.

iii *Multiple traffic processing and response time compliance.* The model should be able to process multiple messages with different response time thresholds and ensure neither of the thresholds is exceeded. For purposes of this research, it is assumed none of the paths in the networks violate the established operating time constraints. The reason for this assumption is that the IEEE test input network represent an actual version of a portion of the power grid. Therefore, it makes sense that this assumption will hold; otherwise the power grid network would mis-operate under normal conditions.

Once each phase was entered into XPress-MP, the next step was to validate that portion of the model entered. In order to do this, the model was run against small networks; starting from three, four, five, seven nodes and a relatively small number of branches as well. These smaller networks were solved manually to make sure that the model was satisfying all the constraints entered in it. When XPress-MP produced different answers, they verified to make sure that there was not another feasible answer that was looked over and that the model had found which was correct as well.

3.5 Response Times or Thresholds

The response times utilized in this research were obtained from Fig. 2.3 and are shown in Table 3.1. This values were entered as input to the model in a separate file containing an array holding these values. Additionally, a second file containing the values in the second row represent experimental values used to test the model against lower time constraints. The resulting configurations were compared between both response time files.

Table 3.1: Time Constraint in milliseconds

	Time in milliseconds											
Time 1	2	4	40	180	300	10000	10000	36000	10000	6	540	2000
Time 2	.2	.7	.25	180	30	10	10	36	100	2	540	1000

3.6 Summary

This chapter discussed the approach taken to solve the problem and the steps followed to execute this research. We started from getting the raw data files and performing calculations on it, to derive information that was needed. It also covered the basis for the optimization technique used on that data and the optimizer software utilized on the data. Next, it described shortly how the mathematical model was built and entered into the program. And finally, it described the process used to make sure that the model was providing correct results before it was used in larger scenarios.

IV. Analysis and Results

This chapter provides details of the results of the optimization trials of our trust node placement algorithm on different network configurations based on standard IEEE test cases. It presents calculations of the best configurations that the Mosel optimizer produced. These resulting configurations are representative of actual systems that the trust system could help to secure in the electric grid. Moreover, this type of optimization model could be utilized to configure any network that has the same characteristics, constraints, and assumptions.

There are a large number of results that are not shown in this document for purposes of brevity; more specifically, the smaller scenarios are excluded. The first set of runs shown here will have the results obtained from running the network in Linux and Windows. Finally, it was not possible to show the larger (i.e., 57 bus) scenarios because their running times were too large given the computing resources available.

4.1 XPress-MP Platforms

The research facility had two different licenses for the XPress-MP optimization software, a Windows version and a Linux version. As a comparison, a section of this chapter will focus on version result differences. The same exact scenario will be compared between Windows and Linux test runs.

4.2 Results analyzed, and questions answered

During this chapter, the analysis will show that it is possible to utilize the trust system suggested in Chapter 2, to enhance system security, despite the trust system delays introduced, when careful optimization constraints are enforced. The results analyzed present suggested configurations that use careful network compartmentalization, which in addition to the security protection in the trust system, also helps to isolate attacks and other malfunctions that cause system

instabilities. Additionally, an evaluation of running times is provided between scenarios.

For purposes of this research, a successful output configuration represents a correct domain grouping in the network which implies that the nodes are connected to each other and that a domain does not contain less nodes than the minimum entered for that run. Also, that the trust systems have been placed at nodes or buses where the security of the domains that they were placed in is increased. Finally, there should be not more than the maximum number of allowed trust nodes placed for the run.

4.3 Input File

As described in Chapter 3, the Mosel model received a text file as input. This file describes the characteristics of a "real life" network, which was assumed to parallel the same nodes and edges as in the raw IEEE test case. This file is the product of the data pre-processing program described in Section 3.4.1, which takes the raw data and to arrive at a corresponding communication network.

The file naming convention used for input files is the following:

$$carlos\_AA\_BB\_CC\_DD\_EE.dat$$

AA= Number of nodes in the network.

BB= Number of branches in the network

CC= Number of messages or path traversal**

DD= Network variation number (used mainly to test the model)

EE= Never used

** For purposes of this research, a message represents a path of buses traversed. The model uses these paths entered to evaluate their total delay and verify that the path has not violated its timing constraint. These paths are part of the input file.

4.4 Result evaluation

The information needed to interpret the result is described in this section.

4.4.1 Output file. The output file generated by the model contained a description of the scenario run and the solution found during after the optimization was completed.

The output included the following information:

1. Scenario ran
2. Threshold file used during the trial
3. Incidence matrices for network as well as for domains created
4. Locations in the network where trust nodes were placed
5. Communication Protection and control traffic entered
6. Delay induced by trust nodes
7. Threshold values used
8. Delay caused by traversing the path
9. Total delay (trust node Delay + Path Delay)

4.4.2 Measurements, Units and Calculations. IEEE raw test case information is provided using the English measurement system (i.e. inches, miles, etc). All the information used in this research was converted to the metric system. The delay unit used for network input was milliseconds (msec). However, the run times produced by the model are given in seconds (secs).

4.4.3 Figure interpretation. The figures in this chapter showing the resulting configurations are interpreted as follows:

- The colored/shaded regions represent the domains formed. And the buses inside this regions belong to this domain.

- The red/dark buses represent a bus where a trust node has been placed.

4.5 Observation variables

The variables being modified on each scenario to observe their effect on the result produced are the following:

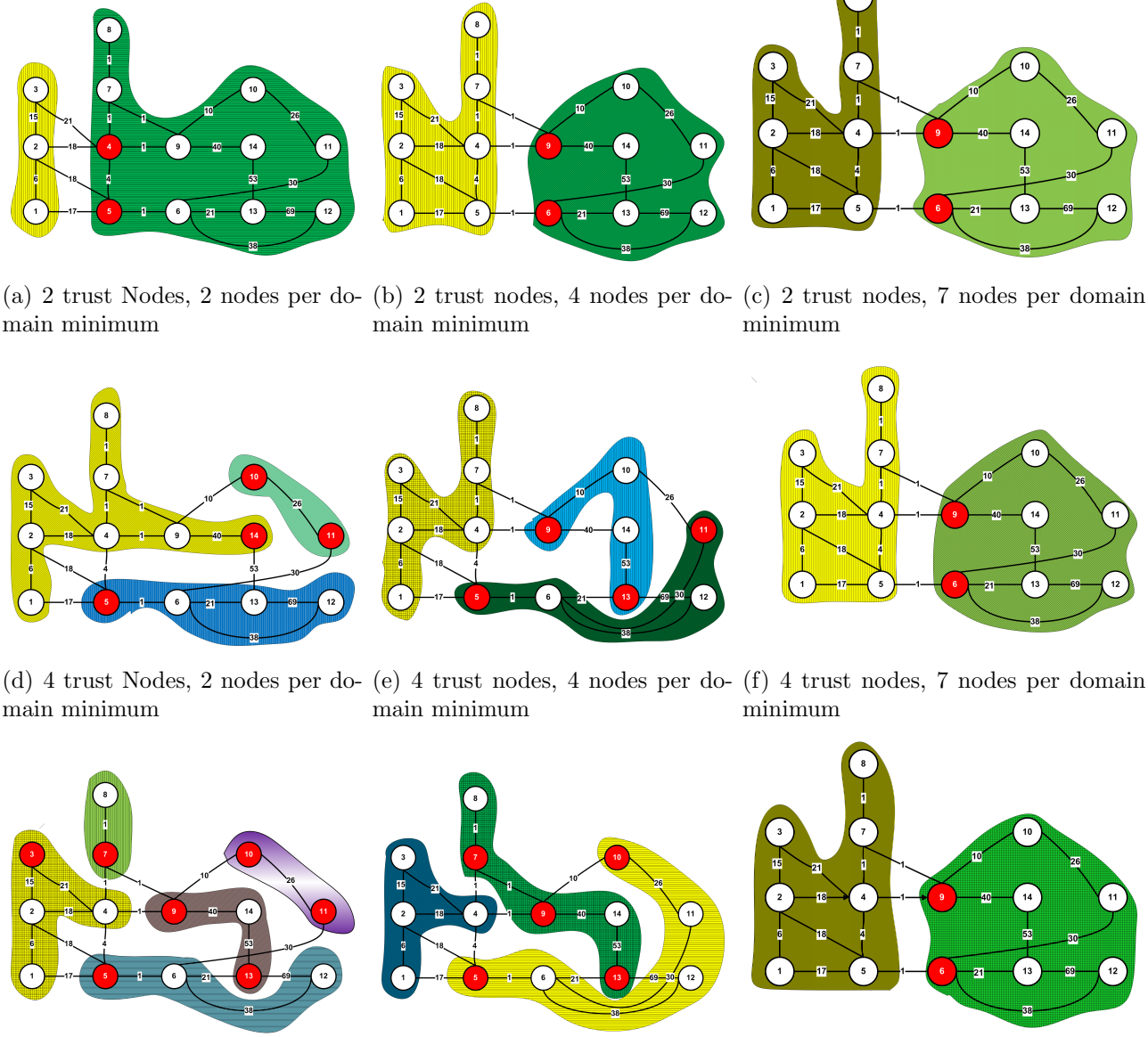
- Minimum number of nodes per domain or sub-network (Section 3.2).
- Maximum number of trust nodes available.

By running scenarios with different value for these variables, it was expected that different configurations would be produced by the model.

We now evaluate individually the effects caused by the variables described in Section 4.5.

4.6 Model Variables Effects

The first variable evaluated in this section is the minimum number of buses on each domain, and the second is the maximum number of trust nodes allowed during that trial. The figure sequence shown in Fig. 4.1 demonstrate the effects caused by the modification of the variables that determine the maximum number of trust nodes and the minimum number of buses per domain. This set of trials were run using the messages shown in five-message section.



(a) 2 trust Nodes, 2 nodes per domain minimum

(b) 2 trust nodes, 4 nodes per domain minimum

(c) 2 trust nodes, 7 nodes per domain minimum

(d) 4 trust Nodes, 2 nodes per domain minimum

(e) 4 trust nodes, 4 nodes per domain minimum

(f) 4 trust nodes, 7 nodes per domain minimum

(g) 7 trust Nodes, 2 nodes per domain minimum

(h) 7 trust nodes, 4 nodes per domain minimum

(i) 7 trust nodes, 7 nodes per domain minimum

Figure 4.1: Configuration changes using different input values on the same scenario

4.6.1 Minimum number of buses per domain. This variable determines the least amount of buses that the optimizer may place in a domain. It is a semi-continuous variable, because it is not allowed to have values greater than zero or less than the minimum value entered. In other words, the domains created can have zero elements or be greater than or equal to the minimum value entered.

The resulting configurations show that by increasing this variable and keeping the other variable constant will have different effects depending on the magnitude of the value of the maximum number of trust nodes. If we observe Fig 4.1 and evaluate the results shown from figure 1(a) through 1(c), we see that when the maximum trust nodes value is small, the effect of the minimum number of buses variable on the result is minimum. This is not the case on the second and third line, it modifies on the configuration produced, the size of the domains increases as the value of this variable increase as well.

4.6.2 Maximum number of trust nodes. We can observe this variable's effect if we examine the series of figures vertically. In the first column of figures, the minimum number of buses remained constant at two as the maximum number of trust nodes increases the number of domains produced increased. The optimizer is able to produce more domains with the more trust nodes it was allowed to work with. However, as we move to the last column its effect is reduced to the point where on the very last figure, even though it is allowed to place seven trust nodes it only utilized two. The minimum number of buses per node did not allow for more domains to be created.

4.6.3 Variable effect analysis. As we have seen this variables may or may not have an effect on the configuration produced. It will depend closely on the value of the other variable. When both variables have a small value, the output will very similar to the output produced when both values have large values. The output will be bounded by the value of the number of trust nodes is small regardless of the value of the number of nodes. On the other hand, the result is bounded by the minimum number of trust nodes, regardless of the number of trust nodes allowed. It is also worth mentioning, that it appears that there is a point in the experiment where the output is similar, if we observe the top row and right column, the configuration is almost identical. However, figures 1(b), 1(e), 1(f), and 1(f) reflect a diversity of topologies.

Table 4.1 shows the running times for the trials shown in Fig.4.1.

Table 4.1: Running times for network with 14 Nodes, and 5 messages in Windows

Sub-figure number	**Trust nodes**	**Min Nodes p/domain**	**Domains created**	**Running Time**
1(a)	2	2	2	8.999
1(b)	2	4	2	0.546
1(c)	2	7	2	0.110
1(d)	4	2	3	48.061
1(e)	4	4	3	0.203
1(f)	4	7	2	0.109
1(g)	7	2	5	25.811
1(h)	7	4	3	0.172
1(i)	7	7	2	0.188

From this table, we can see that the cases 1(d) and 1(g) are the slowest running cases. This is somewhat reasonable, since the number of trust nodes has increased, and with a low minimum number of trust nodes is able to break down the network into smaller domains. Therefore the number of feasible solutions increases increasing the solution space that is being searched as well.

It is worth mentioning that similar behavior was observed in other scenarios ran during this research. We now examine specific cases of the different topologies examined during this research.

4.7 Scenario Runs

There were a total of nineteen different scenarios ran throughout this research. Each scenario has a different number of nodes, edges or messages added to the file. This included three scenarios with four nodes, five scenarios with five nodes, three scenarios with seven nodes, three scenarios with fourteen nodes, one scenario with twenty nodes, two scenarios with thirty nodes, and one scenario with fifty-seven nodes. The scenarios that have the same number of nodes or edges, are different because the nodes are connected differently or the weight of the edges is different.

Each scenario was ran several times (see Table 4.2) using different values for the following variables in the input file:

For brevity, only three scenarios are examined during this chapter. Table 4.2 shows the three scenarios evaluated in this chapter.

Table 4.2: Number of runs per scenario

Scenario	Number of Runs
14 Bus, 20 Branch	75
20 Bus 39 Branch	25
30 Bus 41 Branch	9

4.7.1 Fourteen Node Scenario. This scenario was obtained from the IEEE 14 Bust Test Case, which represent a portion of the American Electric Power System, Midwestern US; as of February, 1962.

This network scenario was run using three different input files with three, five and ten messages (or test paths).

4.7.1.1 ***Three-message scenario, Linux runs.*** This scenario was run in both operating systems, Linux and Windows. Following are some of the times obtained during the runs that were processed in Linux; these scenarios are listed in Table 4.3:

Table 4.3: Running times for Network with 14 Nodes, and 3 messages in Linux

Op System	Trust nodes	Min Nodes p/domain	Domains created	Running Time
Linux	2	5	2	0.108
Linux	3	7	4	0.177
Linux	5	3	4	0.295
Linux	2	3	2	1.641
Linux	7	2	5	616.000
Linux	2	2	2	10.389
Linux	5	2	4	313.000
Linux	5	4	3	0.146
Linux	3	3	2	1.058

The fastest run was the case with two trust nodes and a minimum of five nodes per domain. The model created two domains. We can observe the configuration resulting from the optimization in Fig. 4.2. It is important to observe that the optimizer placed the trust nodes at nodes where messages and any traffic between domains can be monitored. Node five, has three branches leading to the other domain, and node nine has two branches; there are no other branches between domains.

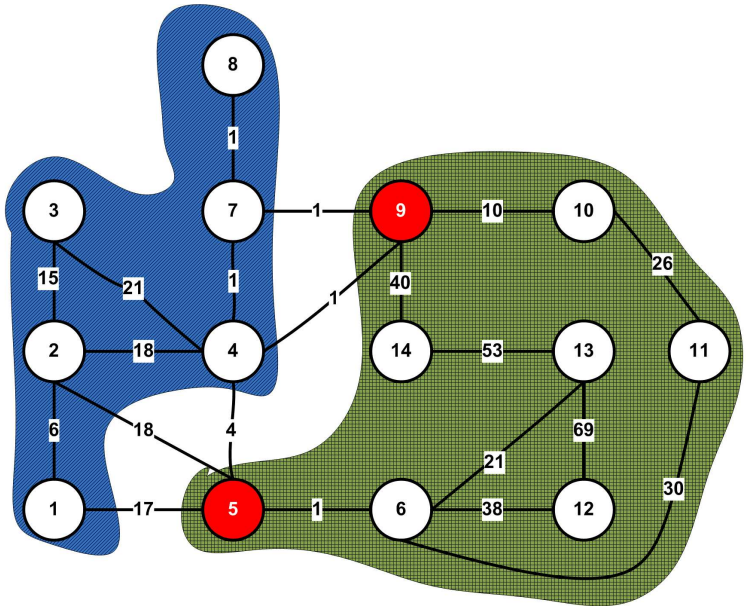


Figure 4.2: Configuration for a 14 Node network, 2 trust nodes, 5 minimum nodes per domain

Table 4.4 shows the paths tested on this scenario and also the delay times caused by the path only, the trust nodes that were placed along the path, and lastly the threshold that paths was subject. The total delay on each of the paths is less than the time threshold they were subject to.

Table 4.4: Message paths for the 14 node, 3 message fastest case in Linux

	Nodes Traversed					Time	tN Delay	Total Delay	Threshold
Path 1	6	13				42	0	42	2000
Path 2	7	4	5	6		12	1200	1212	4000
Path 3	8	7	4	5	12	90	1200	1290	4000

The slowest run was seen in the case where there were seven trust nodes available and a minimum of two nodes per domain allowed. For this case, the optimization allowed five domains to be created. The optimizer utilized all seven trust nodes, and we can observe in Fig. 4.3 that it places the trust nodes in such a way that the communication between domains is secured. Similarly, the minimum number of nodes per domain has been satisfied as well. The trust node requirement is that at least one of the domains monitors the transmission of messages, so that minimize the delay introduced by trust nodes but at the same time, each message traveling between domains is either checked for security irregularities at the time is leaving or arriving to the domain. The resulting configuration in Fig. 4.3 enforces the above description.

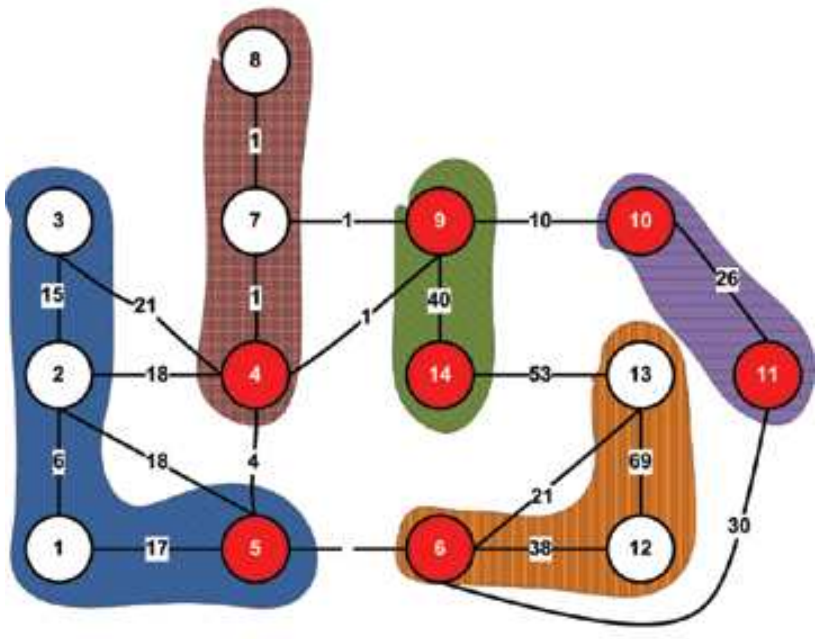


Figure 4.3: Configuration for a 14 Node network, 7 trust nodes, 2 minimum nodes per domain

Table 4.5 show the values of the delays that were induced due to the path traversal and its passing through the nodes with trust nodes. We can observe that message number three approaches the threshold.

Table 4.5: Message paths for the 14 node, 3 message slowest case in Linux

	Nodes Traversed					Time	tN Delay	Total Delay	Threshold
Path 1	1	2	3			21	0	21	2000
Path 2	6	13				21	1200	1221	4000
Path 3	7	4	5	6		6	1800	1806	4000

4.7.1.2 ***Three-message, Windows runs.*** For this scenario, the times were very close to the results obtained with the Linux version. Table 4.6, shows some of the times obtained.

Table 4.6: Running times for Network with 14 Nodes, and 3 messages in Windows

Op System	**Trust nodes**	**Min Nodes p/domain**	**Domains created**	**Running Time**
Windows	2	5	2	0.094
Windows	3	7	2	0.094
Windows	5	3	2	0.930
Windows	2	3	2	1.641
Windows	7	2	5	616.000
Windows	2	2	2	10.827
Windows	5	2	4	312.000
Windows	5	4	3	0.146
Windows	3	3	3	1.656

The fastest scenario was the scenario with two trust nodes and two nodes per domain minimum, its time was 0.094. Fig. 4.4, shows the configuration that the optimizer produced.

This case illustrates what the optimizer opts to do when the number of resources is small. The optimizer is allowed only 2 trust nodes, therefore looks for a better way to satisfy the constraints of contiguous nodes in domains and also trust node strategic placement monitor incoming and outgoing messages. In this case, trust nodes four and five are good solutions. There can be many other solutions to this case. The network configurations output are feasible solutions given the constraints that the input parameters were subject to during the optimization.

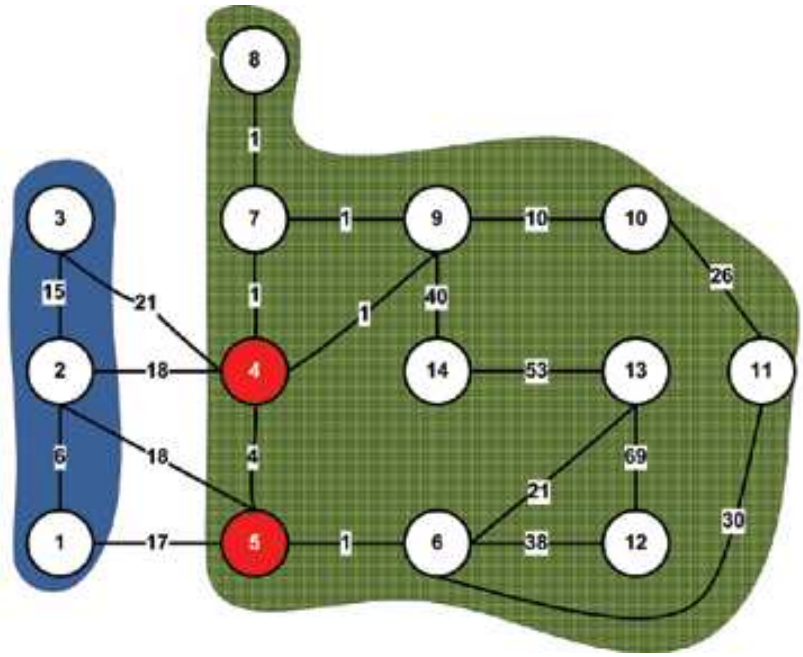


Figure 4.4: Configuration for a 14 Node network, 2 trust nodes, 2 minimum nodes per domain

Table 4.7 shows the delay accumulation of the three messages passed on to the optimizer, neither of them exceeded time constraints. The longest message passes through two trust nodes which adds $1,200$ msecs. of delay, but it is still less than the threshold for messages transmission.

Table 4.7: Message paths traversed for the 14 Node 3 message fastest case in Windows

	Nodes Traversed					Time	tN Delay	Total Delay	Threshold
Path 1	1	2	3			21	0	21	2000
Path 2	6	13				21	0	21	4000
Path 3	7	4	5	6		6	1200	1206	4000

The case with the slowest running time evaluated was run with seven trust nodes and a minimum of two nodes per domain. Fig.4.5 shows the network configuration produced after the trial was run.

Table 4.8 shows the paths that were evaluated during this trial. The path with the longest delay is path three with $1,207$ msecs. The model has arranged the

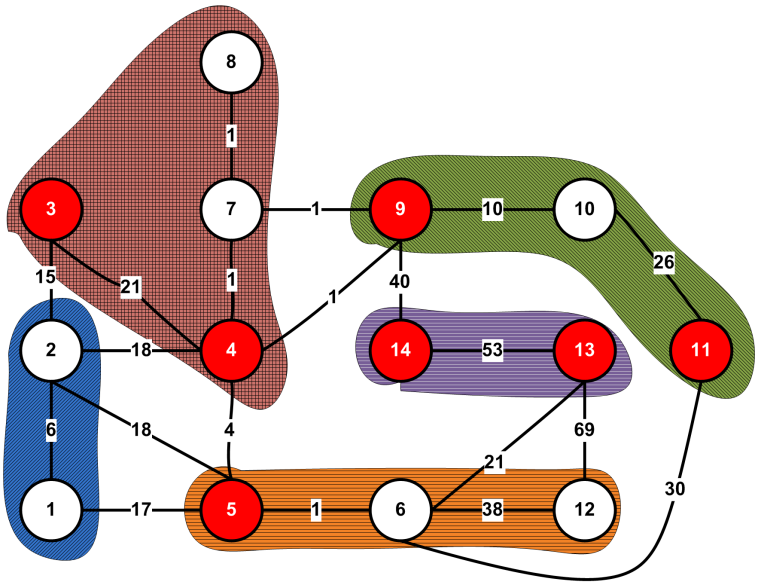


Figure 4.5: Configuration for a 14 Node network, 7 trust nodes, 2 minimum nodes per domain

domains and the locations of the trust nodes considering the traffic of messages input to the model.

Table 4.8: Message paths traversed for the 14 Node 3 message slowest case in Windows

	Nodes Traversed					Time	tN Delay	Total Delay	Threshold
Path 1	1	2	3			21	600	621	2000
Path 2	2	5	6	12		57	600	657	4000
Path 3	8	7	4	5	6	7	1200	1207	4000

4.7.1.3 ***Five-message , Linux runs*** . The results obtained in this trial can be observed in Table 4.9.

The fastest run occurred when the model was allowed a maximum of six trust nodes and the minimum number of nodes per domain was set to five. The resulting running time was 0.15 msecs. Fig. 4.6 contains the solution provided by the optimizer.

Table 4.10 shows the results for this run. The message paths used in this run were entered manually. Consequently, we find some paths that are not the shortest

Table 4.9: Running times for 14 node network with and 5 messages in Linux

Op System	Trust nodes	Min Nodes p/domain	Domains created	Running Time
Linux	10	2	4	23.768
Linux	7	4	3	0.150
Linux	7	3	4	0.238
Linux	7	2	4	11.227
Linux	10	2	4	23.768
Linux	4	2	3	20.408
Linux	4	3	2	4.717
Linux	7	2	5	47.940
Linux	7	3	4	0.238
Linux	7	4	3	0.150
Linux	6	5	2	0.105
Linux	6	3	4	0.310
Linux	6	4	3	0.145
Linux	6	2	5	12.605
Linux	4	2	3	139.000
Linux	5	2	2	9.860

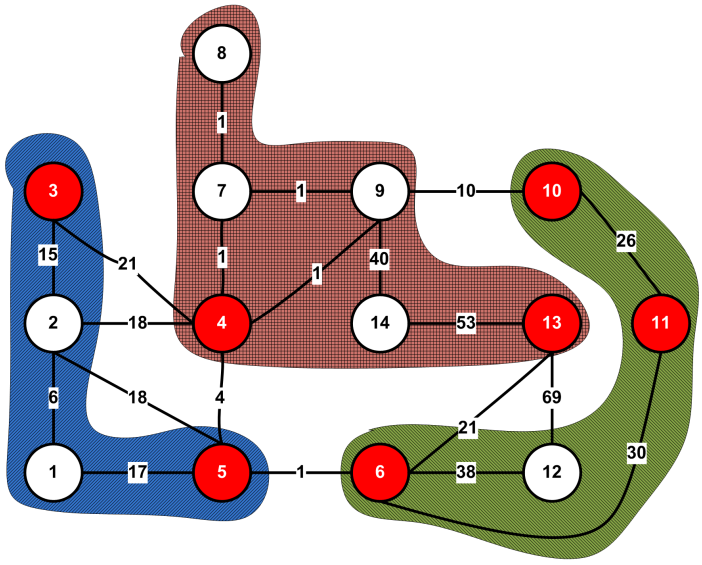


Figure 4.6: Configuration for a 14 Node network, 7 trust nodes, 4 minimum nodes per domain

paths. However, the results still come short from the thresholds that they were run against. The path with the longest delay is path number two. Although the delay caused by the traversing is only 58 msecs, the message stops at 4 trust nodes

causing the delay to quickly increase. To solve this, rules can carefully be defined , so that we do not duplicate security functions within the same domain.

Table 4.10: Message paths and delays for the 14 Node, 5 message fastest case in Linux

	Nodes Traversed						Time	tN Delay	Total Delay	Threshold
Path 1	1	2	4	9	14	13	118	1200	1318	2000
Path 2	11	10	9	4	3		58	2400	2458	4000
Path 3	8	7	9	14	13	12	164	600	764	4000
Path 4	14	9	4	2			59	600	659	180000
Path 5	5	4	7	9	10		16	1800	1816	300000

The slowest run in this scenario happened when the maximum trust node allowed variable was set to 4 and the minimum nodes per domain was set to 2. The optimizer provided 3 domains with a large domain containing nine nodes out of the fourteen; the other 2 domains had three and two nodes in them. The reason, for this number was mainly because of the number of trust nodes that were available. Fig. 4.7 shows the domain configuration mentioned above.

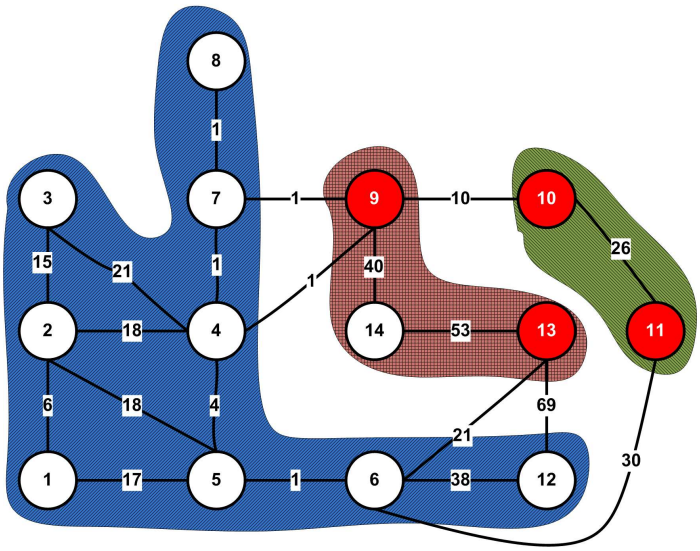


Figure 4.7: Configuration for a 14 Node network, 4 trust nodes, 2 minimum nodes per domain

In Table 4.11 the message paths used in the previous section are used with different configuration. The optimizer outputs configurations which enforce the times constraints or thresholds listed in the last column.

The thresholds of $180,000$ and $300,000$ *msecs* entered correspond to the *thermal overload and poorly damped, or un-damped oscillations* time constraints from the *Backup protection and Control; WAPaC* system. The threshold do not state a specific length of time, so an estimate of 180 *secs* and 300 *secs* was entered. For reference see table 2.3.

Table 4.11: Message paths traversed for the 14 Node, 5 message fastest case in Linux

	Nodes Traversed						Time	tN Delay	Total Delay	Threshold
Path 1	1	2	4	9	14	13	118	1200	1318	2000
Path 2	11	10	9	4	3		58	1800	2458	4000
Path 3	8	7	9	14	13	12	164	1200	764	4000
Path 4	14	9	4	2			59	600	659	180000
Path 5	5	4	7	9	10		16	1200	1816	300000

4.7.1.4 ***Five-message, Windows runs.*** Fig. 4.12 displays the results for the 14 node case. There is no significant difference between the Windows cases.

The case with ten trust nodes and a minimum of two nodes per domains is shown in Fig. 4.8. The model produced a total of four domains, and placed the trust nodes in nodes five, six, seven, nine, ten, eleven, and twelve.

The model choses a solution where it has grouped a set of nodes based on the degree of the protection that the placement of a trust node adds to the network, without regarding proximity. For example, node nine and fourteen have a delay of $40msecs$ and the edge between nine and four, or seven has a delay of $1msec$ and still nodes nine and fourteen have been placed together but separate from nodes four and seven. This is because the constraints allow to place greater importance when choosing the position of a trust node.

Table 4.12: Running times for scenario with 14 Nodes, 5 messages in Windows

Op System	Trust nodes	Min Nodes p/domain	Domains created	Running Time
Windows	10	2	5	10.64
Windows	7	4	3	0.187
Windows	7	3	4	0.391
Windows	7	2	5	25.186
Windows	6	5	2	0.109
Windows	6	4	3	0.344
Windows	6	3	4	0.500
Windows	6	2	5	7.672
Windows	5	2	4	135.00
Windows	4	3	3	1.328
Windows	4	2	3	48.061
Windows	3	2	3	36.545
Windows	2	2	2	8.968

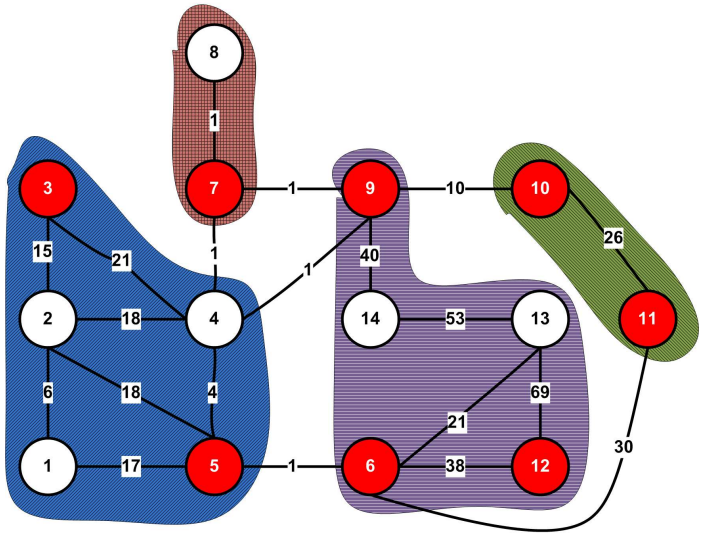


Figure 4.8: 14 Node network, Maximum of 10 trust nodes, and a minimum of 2 nodes per domain

Table 4.13 shows the times resulting from this run. The message paths input, were entered manually and do not represent the shortest path between the source and destination node.

Next, Fig. 4.9 displays what the optimizer does as the number of trust nodes allowed is reduced. It creates larger domains so that it compensates for the reduction

Table 4.13: Message paths traversed for the 14 Node, 5 message fastest case in Windows

	Nodes Traversed						Time	tN Delay	Total Delay	Threshold
Path 1	1	2	4	9	14	13	118	600	718	2000
Path 2	11	10	9	4	3		58	2400	2458	4000
Path 3	8	7	9	14	13	12	164	1800	1964	4000
Path 4	14	9	4	2			59	600	659	180000
Path 5	5	4	7	9	10		16	2400	2416	300000

in trust nodes. This way, the need for trust nodes is reduced because the branches between domains are less and it can use the fewer trust nodes more effectively.

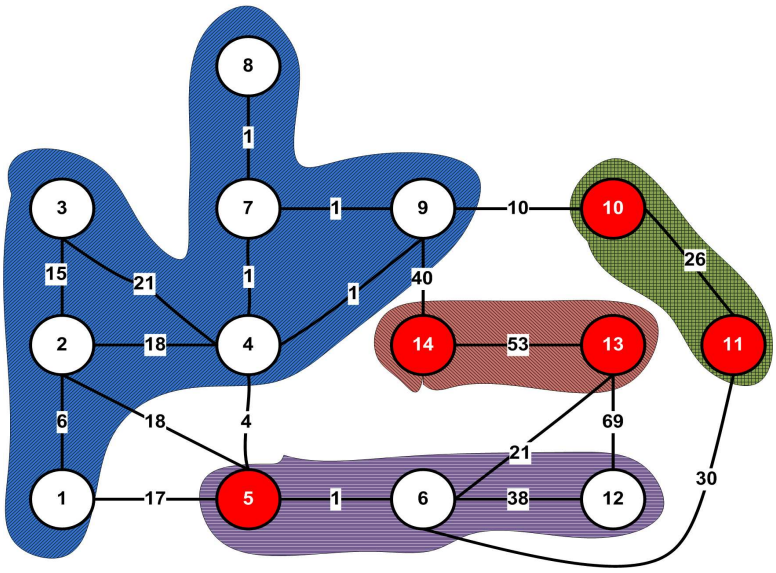


Figure 4.9: 14 Node network, Maximum of 5 trust nodes, and a minimum of 2 nodes per domain

Table 4.14 shows the paths that were tested. We can observe that path number two gets closer to the time constraint corresponding to its type of message, however, the output configuration kept it below the time constraint, although the optimizer utilized all of the trust nodes, it was able to placed them such that the message traffic input did not exceed the constraints.

The minimum running time for this scenario was $0.105 msecs$, while the maximum was $139.00 msecs$ and the average was $21.148 msecs$. There is no noticeable difference between the Windows and Linux runs of this scenario.

Table 4.14: Message paths traversed for the 14 Node 5 message case in Windows

	Nodes Traversed						Time	tN Delay	Total Delay	Threshold
Path 1	1	2	4	9	14	13	118	1200	1318	2000
Path 2	11	10	9	4	3		58	1200	1258	4000
Path 3	8	7	9	14	13	12	164	1200	1364	4000
Path 4	1	2	4	7	8		26	0	26	2000
Path 5	6	5	4	9	10		16	1200	1216	300000

4.7.1.5 ***Ten message Linux runs.*** The results for this case are shown in Table 4.15. There is a climb in the running times for this scenario. The number of messages appear to start having an effect on the complexity of the optimization.

Table 4.15: Running times for Network with 14 Nodes, and 5 messages in Linux

Op System	**Trust nodes**	**Min Nodes p/domain**	**Domains created**	**Running Time**
Linux	10	2	6	357.000
Linux	7	4	3	0.170
Linux	7	3	4	0.188
Linux	7	2	5	3640.000
Linux	6	5	2	0.134
Linux	6	4	3	0.170
Linux	6	3	4	0.720
Linux	6	2	5	10844.000
Linux	5	2	4	16469.000
Linux	4	3	3	35.944
Linux	4	2	2	0.762
Linux	3	3	3	3.103
Linux	3	2	3	77.000
Linux	2	2	2	7.585

As we can see, if the number of trust nodes is left constant and reduce the minimum number of nodes per domain the running times increase rapidly. However, as the maximum number of trust nodes is reduced as well, the rate at which the running times increases slows down.

The overall statistics show a spike in the running times. The fastest scenario ran for 0.134 secs, and the slowest scenarios ran for $16,469$ secs. The average time was $2,336.052$ secs. This number shows that the time spread in the data points is large.

Fig. 4.10 represents the configuration produced by the optimizer. Note that the domain composed by nodes 1,2,3,4,5 does not contain a trust node in it. Nonetheless, overall network security has been preserved. However, there is no branch leaving this domain that connects to other domains where a trust node is not present.

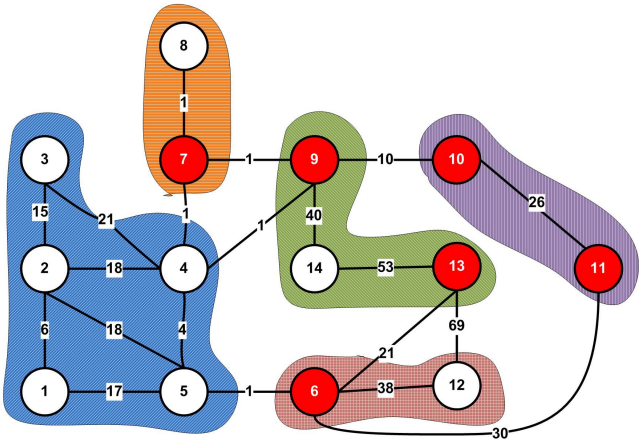


Figure 4.10: Configuration for a 14 Node network, 6 trust nodes, 2 minimum nodes per domain

Table 4.16 shows the resulting delay for all ten messages entered for this trial. Each simulated message has been assigned a different message type from the operating constraints defined in Chapter 2. We can see that every message has met the time constraint that it has been subject to.

Table 4.16: Message paths traversed for the 14 Node, 10 message case in Linux

	Nodes Traversed					Time	tN Delay	Total Delay	Threshold
Path 1	4	9	10			11	1200	1211	2000
Path 2	1	5	4	9	14	62	600	662	4000
Path 3	14	9	7			41	1200	1241	4000
Path 4	2	4				18	0	18	100000
Path 5	8	7	9			2	1200	1202	180000
Path 6	7	4	5	6	12	44	1200	1244	300000
Path 7	11	6	5	2		49	1200	1249	6000
Path 8	3	4	5	6	12	64	600	664	3600000
Path 9	5	6	12			39	600	639	540000
Path 10	13	6	5	4	3	47	1200	1247	100000

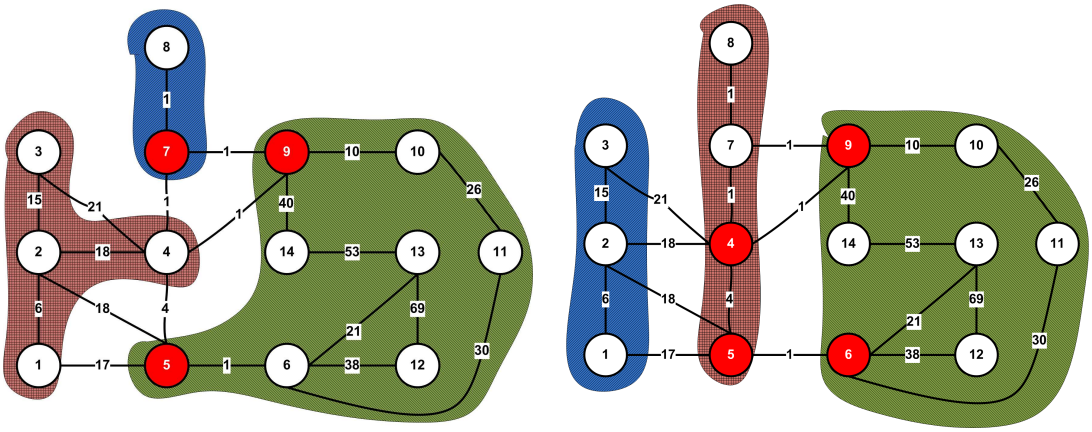
4.7.1.6 ***Ten message Windows runs.*** Table 4.17 shows the running times obtained during this case.

Table 4.17: Running times for Network with 14 Nodes, and 10 messages in Windows

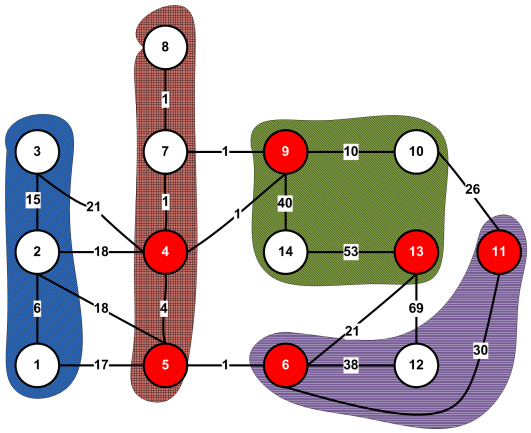
Op System	**Trust nodes**	**Min Nodes p/domain**	**Domains created**	**Running Time**
Windows	10	2	6	916.000
Windows	7	4	3	0.171
Windows	7	3	4	0.219
Windows	7	2	5	3640.000
Windows	6	5	2	0.125
Windows	6	4	3	0.375
Windows	6	3	4	0.625
Windows	6	2	5	3329.000
Windows	5	2	-	CRASHED
Windows	4	3	3	14.417
Windows	4	2	3	759.000
Windows	3	2	3	74.000
Windows	2	2	2	8.559

The next three figures provide an idea of how the optimizer behaves when the values of the test variables of maximum amount of trust nodes and the minimum number of nodes per domain are changed. The figures illustrate the evolution of modifications performed as the parameters change. We can see starting from Fig.11(a) how the solution evolves to Fig.11(b) and ends with Fig.11(c). As the

number of trust nodes increases the optimizer is able to build solutions with more domains, because it is able to protect the traffic between them as long as it does not go below the minimum number of nodes per domain.



(a) 3 trust Nodes, 2 nodes per domain minimum (b) 4 trust nodes, 3 nodes per domain minimum



(c) 6 trust nodes, 3 nodes per domain minimum

Figure 4.11: Configuration changes using different input values on the same scenario

Table 4.18 shows the delay resulting from running the trial shown on Fig. 11(b). Lastly, the configuration output by the model preserves all the thresholds that have been provided to it.

Table 4.18: Message paths traversed for the 14 Node, 10 message case in Windows

	Nodes Traversed					Time	tN Delay	Total Delay	Threshold
Path 1	4	9	10			11	1200	1211	2000
Path 2	1	5	4	9	14	62	1800	1862	4000
Path 3	14	9	7			41	600	641	4000
Path 4	2	4				18	600	618	100000
Path 5	8	7	9			2	600	602	180000
Path 6	7	4	5	6	12	44	1800	1844	300000
Path 7	11	6	5	2		49	1200	1249	6000
Path 8	3	4	5	6	12	64	1800	1864	3600000
Path 9	5	6	12			39	1200	1239	540000
Path 10	13	6	5	4	3	47	1800	1847	100000

4.7.2 ***Twenty Node Scenario*** . The network represented in this scenario does not represent an actual case of a power grid region. It is a simulated configuration created only to provide a step before moving into a higher node network.

The scenarios reported in this section are only two of the four different network files tested. The first scenario has a total of ten messages, and the second scenario has 20 messages. This scenario was ran in Linux and Windows operating systems. Each one, was ran utilizing twelve different combinations of values for the maximum number of trust nodes and the minimum number of nodes per domain variables.

4.7.2.1 ***Ten message Linux runs.*** The summary of results for this run are shown in Table 4.19.

The trial with 14 trust nodes and 2 minimum nodes per domain shows with great detail what the optimizer does when it has many trust nodes to subdivide the network. It shows that having a large amount of trust nodes increases the security of the network because it allows the optimizer to create a larger number of domains and isolates them by placing a trust node at strategic locations as shown in Fig. 4.12.

Table 4.19: Running times for Network with 20 Nodes, and 10 messages in Linux

Op System	Trust nodes	Min Nodes p/domain	Domains created	Running Time
Linux	14	6	3	1.141
Linux	14	4	5	2.316
Linux	14	3	6	2.776
Linux	14	2	9	3605.000
Linux	10	5	4	2.401
Linux	10	4	5	2.848
Linux	10	3	6	25.130
Linux	10	2	6	3604.000
Linux	5	6	3	0.688
Linux	5	4	3	918.000
Linux	5	3	3	3681.000
Linux	3	3	2	151.000

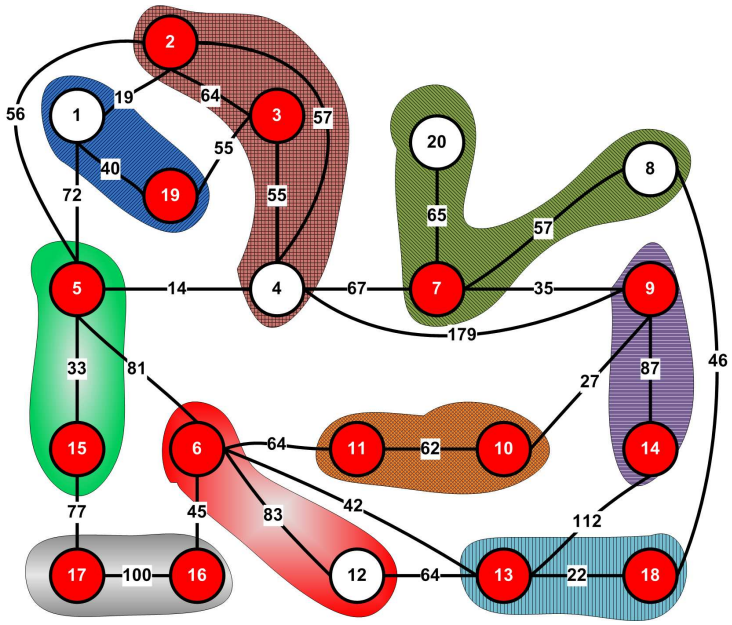


Figure 4.12: 20 Node network, Maximum of 14 trust nodes, and a minimum of 2 nodes per domain

In comparison with the runs from the previous section, there is no noticeable increase of running times. The minimum time is 0.688 secs, the maximum time is 3,681 msecs and the average is 999.6917 secs. The values presented in Table 4.20 support the time constraint premise, since none of the times violate the constraint the bounds the time alloted to that type of message.

Table 4.20: Message paths traversed for the 20 Node, 10 message case in Linux

	Nodes Traversed					Time	tN Delay	Total Delay	Threshold
Path 1	1	19				40	600	640	2000
Path 2	2	4	7	9	10	186	2400	2586	4000
Path 3	3	4	7	9	14	244	2400	2644	4000
Path 4	17	16	6			145	1800	1945	1000000
Path 5	8	18				46	600	646	6000
Path 6	5	4	7	9		116	1800	1916	3600000
Path 7	12	6	5	4	3	233	1800	2033	540000
Path 8	10	11				62	1200	1262	3600000
Path 9	14	13	6	16	17	299	3000	3299	1000000
Path 10	18	13	6	5	2	201	3000	3201	300000

4.7.2.2 ***Ten message, Windows runs.*** The summary of results for this run are shown in Table 4.21

Table 4.21: Running times for Network with 20 Nodes, and 10 messages in Windows

Op System	**Trust nodes**	**Min Nodes p/domain**	**Domains created**	**Running Time**
Windows	10	2	6	916.000
Windows	7	4	3	0.171
Windows	7	3	4	0.219
Windows	7	2	5	3640.000
Windows	6	5	2	0.125
Windows	6	4	3	0.375
Windows	6	3	4	0.625
Windows	6	2	5	3329.000
Windows	5	2	-	CRASHED
Windows	4	3	3	14.417
Windows	4	2	3	759.000
Windows	3	2	3	74.000
Windows	2	2	2	8.559

Fig.4.13 shows the configuration produced when running the model using variables maximum trust nodes of ten, and a minimum nodes per domain of four. The optimizer build a total of five domains, and once again we can observe that the trust nodes have been placed at places where they are able to monitor or oversee the traffic leaving or arriving to that specific domain. If we notice, nodes six and eleven

contain trust nodes and they are contiguous nodes. This situation increases the delay of a message path, and it may be somewhat redundant. This delay could be reduced if we could carefully implement a security system where a trust node is able recognize that a message has been scanned by another trust nodes next to it, or in the same domain and maybe perform basic checks for protection or simply not touch the message.

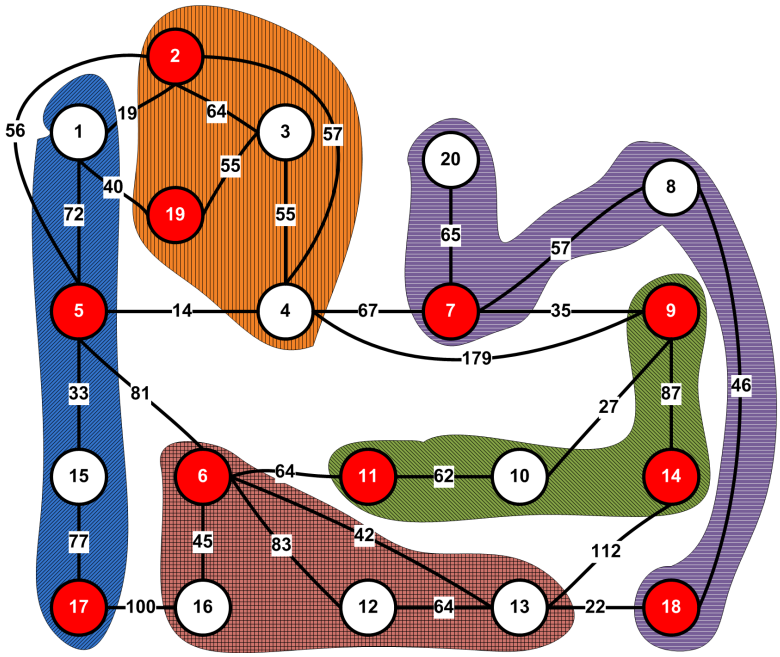


Figure 4.13: 20 Node network, Maximum of 10 trust nodes, and a minimum of 4 nodes per domain

Table 4.22 shows the results obtained from this trial.

Table 4.22: Message paths traversed for the 20 Node, 10 message case in Windows

	Nodes Traversed					Time	tN Delay	Total Delay	Threshold
Path 1	1	19				40	600	640	2000
Path 2	2	4	7	9	10	186	1800	1986	4000
Path 3	3	4	7	9	14	244	1800	2044	4000
Path 4	17	16	6			145	1200	1345	1000000
Path 5	8	18				46	600	646	6000
Path 6	5	4	7	9		116	1800	1916	3600000
Path 7	12	6	5	4	3	233	1200	1433	540000
Path 8	10	11				62	600	662	3600000
Path 9	14	13	6	16	17	299	1800	2099	1000000
Path 10	18	13	6	5	2	201	2400	2601	300000

4.7.2.3 ***Twenty message Linux runs.*** Table 4.23 shows the number of domains formed and the running times for the twenty message Linux runs. We begin to observe a rise in the running times in comparison with previous runs. The longest run was for 2029 secs. for the configuration that allowed five trust nodes and a minimum of three nodes per domain. The fastest trial run was the configuration that had a fourteen trust nodes and a minimum of six domains with a running time of 1.173 secs. The optimizer created three domains to come out with this solution.

Table 4.23: Running times for a 20 Node Network and 20 messages in Linux

Op System	**Trust nodes**	**Min Nodes p/domain**	**Domains created**	**Running Time**
Linux	14	6	3	1.173
Linux	14	6	4	15.269
Linux	14	3	5	58.147
Linux	14	2	3	21699.000
Linux	10	5	4	1.554
Linux	10	4	4	21.412
Linux	10	3	5	54.565
Linux	10	2	6	21634.000
Linux	5	6	3	1.861
Linux	5	4	3	88.000
Linux	5	3	3	2029.000
Linux	3	3	2	66.000

Table 4.24: Message paths traversed for the 20 Node, 20 message case in Linux

	Nodes Traversed						Time	tN Delay	Total Delay	Threshold
Path 1	1	2	4				76	0	76	3600000
Path 2	20	7	8	18	13		190	1800	1990	1000000
Path 3	12	6	5	2			220	1200	1420	6000
Path 4	16	6	13	18	8		155	3000	3155	180000
Path 5	7	8	18				103	1200	1303	540000
Path 6	13	6	16				87	1800	1887	4000
Path 7	17	15	5	4	7	20	256	600	856	2000
Path 8	15	5	4	7	9		149	600	749	300000
Path 9	10	9	7	4	2		186	600	786	3600000
Path 10	8	7	4	5			138	1200	1338	180000
Path 11	1	5	6	13	18		217	2400	2617	6000
Path 12	5	4	7	9	14		203	1200	1403	1000000
Path 13	6	13	18	8			110	2400	2510	300000
Path 14	6	13	18				64	1800	1864	1000000
Path 15	12	6	5	2			220	1200	1420	540000
Path 16	5	4	3				69	600	669	1000000
Path 17	4	5	6	13	18		159	2400	2559	6000
Path 18	13	6	5	2			179	1800	1979	300000
Path 19	12	13	18	8	7	20	254	1800	2054	3600000
Path 20	19	1	5	6	13	18	257	3000	3257	1000000

Fig.4.14 shows the resulting configuration when the system has ten trust nodes available to place in the network. This solution produced a total of six domains. It seems like the answer is not the best solution possible. It contains a domain containing eight nodes and four domains with only two nodes in them. There are several nodes where redundant placement of trust node has occurred. A solution with a more even number of nodes per domain seems to be more appropriate and probably a better utilization of trust node resources to improve network security.

Table 4.24 on page 100, shows the paths entered in to this case. This paths were created automatically using the shortest path application developed for this research.

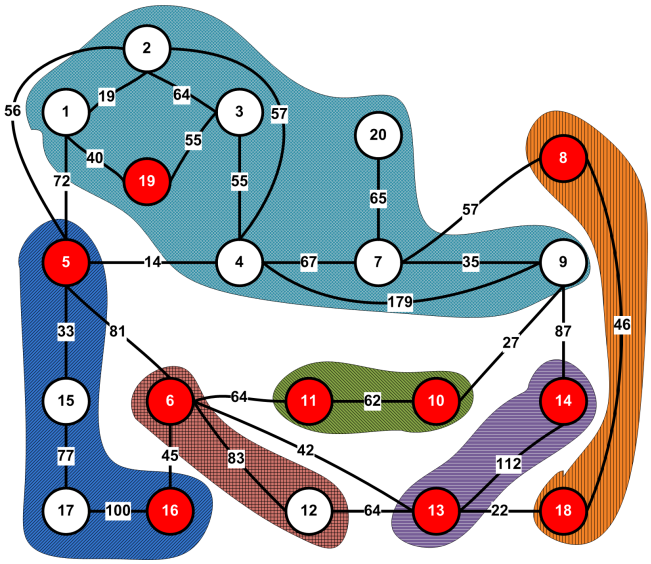


Figure 4.14: 20 Node network, maximum of 10 trust nodes, and a minimum of 2 nodes per domain

4.7.2.4 ***Twenty message Windows runs.*** Table 4.25 on page 102 presents the running times obtained after running this scenario.

The Windows optimizer crashed twice in the first run. Each of the scenarios that crashed were reran for a total of three tries. On all three runs, the *memory available* counter in the IVE had about 760 MBytes left by the time the optimizer halted. Once this happened, the XPress-MP had to be restarted as it would become completely unstable. Therefore, no results were available for those scenarios.

Fig.4.15 on page 102, shows the configuration produced by the windows optimizer for this case. In this case, we increased the minimum number of nodes per domain allowed. This parameter combination seem to produced a better result as far as the number of nodes per domain. The range in the number of nodes per domain is smaller. The most populated domain has six nodes and the least populated has four nodes.

Table 4.25: Running times for Network with 20 Nodes, and 20 messages in Windows

Op System	Trust nodes	Min Nodes p/domain	Domains created	Running Time
Windows	14	6	3	0.547
Windows	14	6	4	6.578
Windows	14	3	5	39.343
Windows	14	2	-	CRASHED
Windows	10	5	4	1.218
Windows	10	4	4	4.406
Windows	10	3	5	42.342
Windows	10	2	-	CRASHED
Windows	5	6	3	1.172
Windows	5	4	3	53.202
Windows	5	3	3	752.000
Windows	3	3	2	74.000

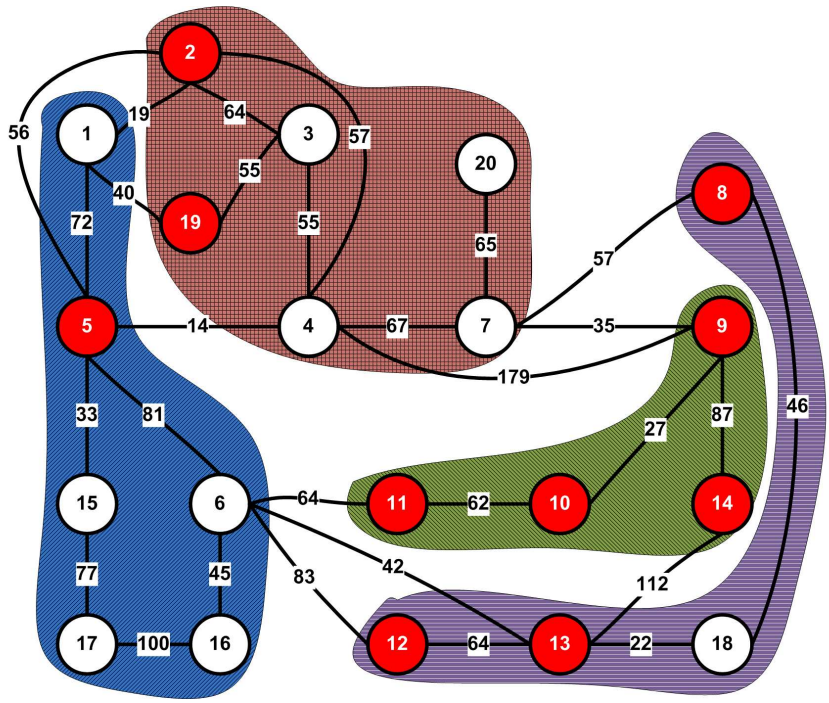


Figure 4.15: 20 Node network, maximum of 10 trust nodes, and a minimum of 4 nodes per domain

Table 4.26 on page 103, shows the delay on each of the paths input to the model, neither of the message paths violated its time constraint. We can see that some of the thresholds are large enough to not allow this. However, we can see that

some of the messages could have violated a 2 msec (2000) constraint if they have had been entered with a different message type.

Table 4.26: Message paths traversed for the 20 Node, 20 message case in Windows

	Nodes Traversed						Time	tN Delay	Total Delay	Threshold
Path 1	1	2	4				76	600	676	3600000
Path 2	20	7	8	18	13		190	1200	1390	1000000
Path 3	12	6	5	2			220	1800	2020	6000
Path 4	16	6	13	18	8		155	1200	1355	180000
Path 5	7	8	18				103	600	703	540000
Path 6	13	6	16				87	600	687	4000
Path 7	17	15	5	4	7	20	256	600	856	2000
Path 8	15	5	4	7	9		149	1200	1349	300000
Path 9	10	9	7	4	2		186	1800	1986	3600000
Path 10	8	7	4	5			138	1200	1338	180000
Path 11	1	5	6	13	18		217	1200	1417	6000
Path 12	5	4	7	9	14		203	1800	2003	1000000
Path 13	6	13	18	8			110	1200	1310	300000
Path 14	6	13	18				64	600	664	1000000
Path 15	12	6	5	2			221	1800	2021	540000
Path 16	5	4	3				69	600	669	1000000
Path 17	4	5	6	13	18		159	1200	1359	6000
Path 18	13	6	5	2			179	1800	1979	300000
Path 19	12	13	18	8	7	20	254	1800	2054	3600000
Path 20	19	1	5	6	13	18	257	1800	2057	1000000

4.7.3 ***Thirty Node Scenario*** . This scenario was built using the data from the Power Systems Test Case Archive. This section shows the different configurations tested with ten messages as the input traffic. Additionally, the results from the Windows and Linux trials are shown here.

4.7.3.1 ***Thirty message Linux runs.*** This section will summarize the results obtained from running the thirty node network varying the values for the maximum number of trust nodes and the minimum number of nodes per domain. Table 4.27 summarizes the running times and the number of nodes created on each

Table 4.27: Running times for 30 Node network with 14 Nodes, and 10 messages in Linux

Op System	Trust nodes	Min Nodes p/domain	Domains created	Running Time
Linux	14	6	5	5.8121
Linux	10	6	5	24.421
Linux	5	6	3	3612.000
Linux	14	4	7	11.966
Linux	10	4	6	3607.000
Linux	5	4	3	3622.000
Linux	15	6	5	6.383
Linux	15	4	7	12.402
Linux	15	8	3	2.405
Linux	12	8	3	2.952
Linux	10	8	3	2.699
Linux	6	8	3	2.995

trial. The configuration that has the shortest running time is the ninth trial in the table, it created three domains and ran for 2.405.

The slowest trial ran for 3622 *secs*. This trial does present a slight increase of running times. If we compare the fastest case from the previous Linux run containing 10 messages (20 Node scenario); we get an increase of over 300% which is probably our greatest increase rate. However, the longest time and the average do not show the same increase rate.

Fig.4.16 illustrates the configuration produced when running the model with ten trust nodes and minimum of six nodes per domain this network with a maximum of ten trust nodes and a minimum of six nodes per domain. The number of nodes per domain is evenly distributed among all the domains. We can observe that the trust nodes have been placed at nodes that connect to other domains, such that no message be able to leave or enter a domain without being monitored.

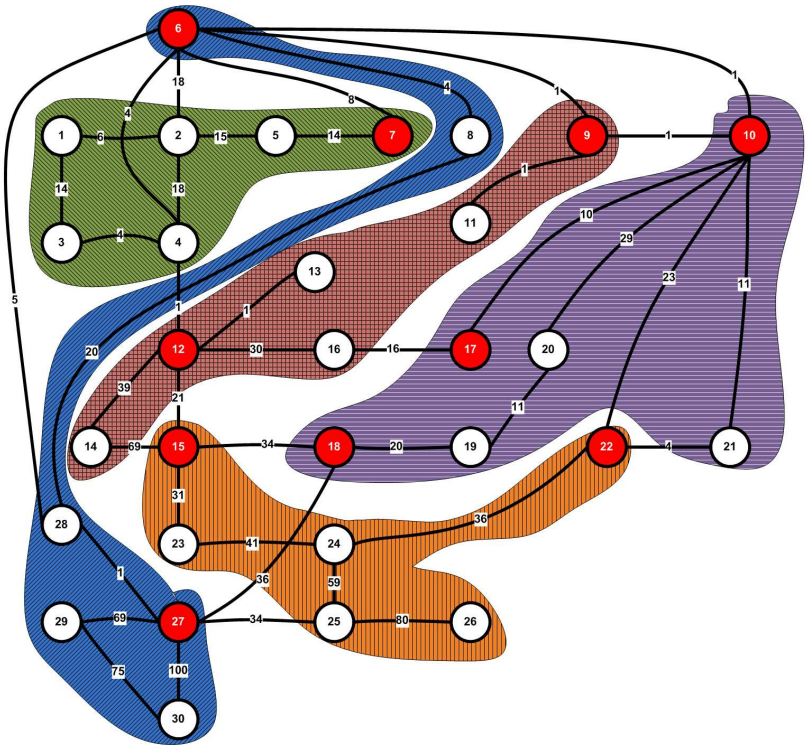


Figure 4.16: 30 Node network, Maximum of 10 trust nodes, and a minimum of 6 nodes per domain

Finally, Table 4.28 summarizes the paths input to the model and their delays with the configuration presented in Fig. 4.16.

Table 4.28: Message paths traversed for the 30 Node, 10 message case in Windows

	Nodes Traversed							Time	tN Delay	Total Delay	Threshold
Path 1	4	6	10					5	1200	1205	2000
Path 2	1	3	4	12	15			40	1200	1240	4000
Path 3	30	27	28	6	7			113	1800	1913	6000
Path 4	22	21	10	6	4	12	14	60	2400	2460	360000
Path 5	18	15	12	4	2			74	1800	1874	4000
Path 6	7	6	9					9	1800	1809	300000
Path 7	6	28						7	600	607	180000
Path 8	11	9	10	21				13	1200	1213	4000
Path 9	12	13						1	600	601	1000000
Path 10	13	12	4	6	10	17		17	2400	2417	540000

4.8 57 Node Scenario

As stated at the beginning of this chapter, the test cases examined in this chapter were not the only ones that were explored. A larger IEEE test case was run on four different occasions. The first two times the scenario was run, it was left to run without a time limit. The first time it ran for approximately over (no record of stopping time was available) three days. After the third day, the run crashed while unattended; there was no error message or anything to provide any information about the cause of the failure. The second time, it ran for four days also. This time, we used a Linux application called *"screen"* which is used to keep applications running under the *"screen"* active even in the event when the connection to the Linux service is dropped or session log offs was used. The process ID corresponding to the Mosel run was monitored closely to identify any circumstances that may have lead to the first failure. The CPU was being used at 100% and the memory was being used at 99.7% as well. Unfortunately, the application failed to keep the session active and no record of this run was recover either.

The last two runs, were timed to run for four and six hours. However, the results that were obtained on both cases were different from each other and also erroneous. The domains created were not contiguous, in other words a domain contains nodes without an edge connecting them different regions within the network but different on both runs. This results suggest the possibility that the run had not arrived to a feasible solutions when it is stopped and that more run time may be needed to achieve a better solution. Appendix VI illustrates one of the configurations output by one of the runs.

4.9 Note on Windows runs

It is important to note, that when the model was executed in Windows; it crashed in several occasions. Even on smaller cases that lasted three or four hours in the Linux version, the Windows based would not be able to complete the trial and crashed. In addition to this, the optimizer would prompt the user to close the

XPress-MP because the system was unstable, after closing the message box the optimizer was noticeable erratic in behavior. The only way to fix this behavior was to close the optimizer completely and restart again.

4.10 Totals

Table 4.29 shows the totals for each of the scenarios ran and final values for the overall research runs. The scenarios that have an asterisk (*) in the average time have trials that crashed and no running time was collected.

Table 4.29: Total Results for the over all runs shown in this document

Scenario	Op System	Fastest	Slowest	Avg Time
14 Node 3 Msgs	Linux	0.099	616.000	104.757
14 Node 3 Msgs	Windows	0.094	1709.000	188.111
14 Node 5 Msgs	Linux	0.105	139.000	17.878
14 Node 5 Msgs	Windows	0.109	135.000	21.149
14 Node 10 Msgs	Linux	0.134	16469.000	2415.386
14 Node 10 Msgs	Windows	0.125	3640.000	*625.186
20 Node 10 Msgs	Linux	0.688	3681.000	999.691
20 Node 10 Msgs	Windows	1.36	1497.000	175.177
20 Node 20 Msgs	Linux	0.956	21634.000	2179.161
20 Node 20 Msgs	Windows	0.547	752.000	*97.481
30 Node 10 Msgs	Linux	2.405	3622.000	909.420
30 Node 10 Msgs	Windows	1.453	116.000	*21.545

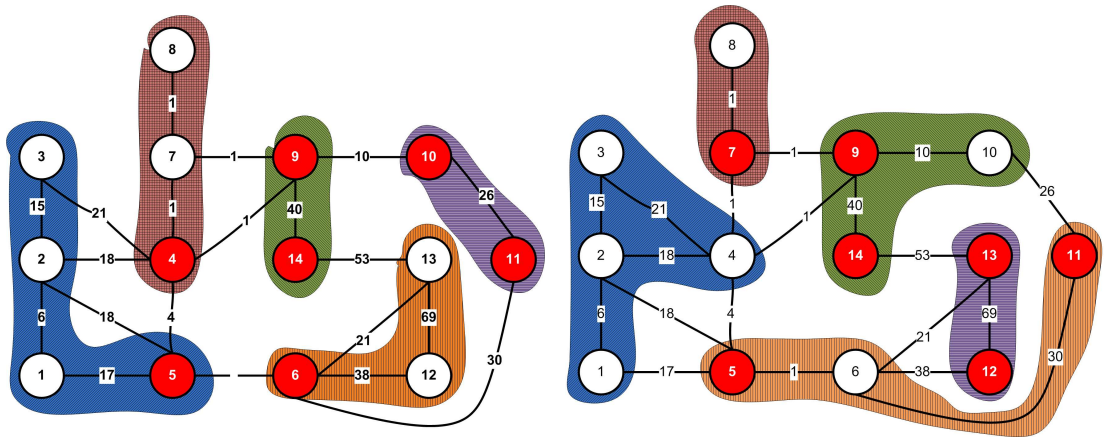
4.11 Reduced Response Times or Thresholds

As an exploration for this research, different scenarios were run utilizing an alternate threshold file containing reduced thresholds to observe how the optimizer changed the output as some of the messages actually exceeded the modified threshold.

It was observed that when the input contains messages (or traffic) that exceeds its threshold, the optimizer generates a different configuration that complies with the traffic needs. In doing so, the model ensures that the traffic patterns entered the

removes trust nodes from the configuration in order to reduce the delay introduced by the trust node throughout the message's path.

To better illustrate this claim, the Fig. 4.17 shows the change caused by the reduction in response time threshold. The input parameters were exactly the same.



(a) Original response threshold times (b) Reduced response threshold times

Figure 4.17: Configurations results using 2 different response time values

The domains formed are different, and therefore the locations of the trust nodes is different as well. Table 4.30 shows the original results.

Table 4.30: Message paths traversed for the 14 Node 3 message slowest case in Windows

	Nodes Traversed					Time	tN Delay	Total Delay	Threshold
Path 1	1	2	4			24	600	624	2000
Path 2	2	5	6			57	1200	1257	4000
Path 3	8	7	4	5	6	7	1800	1807	4000

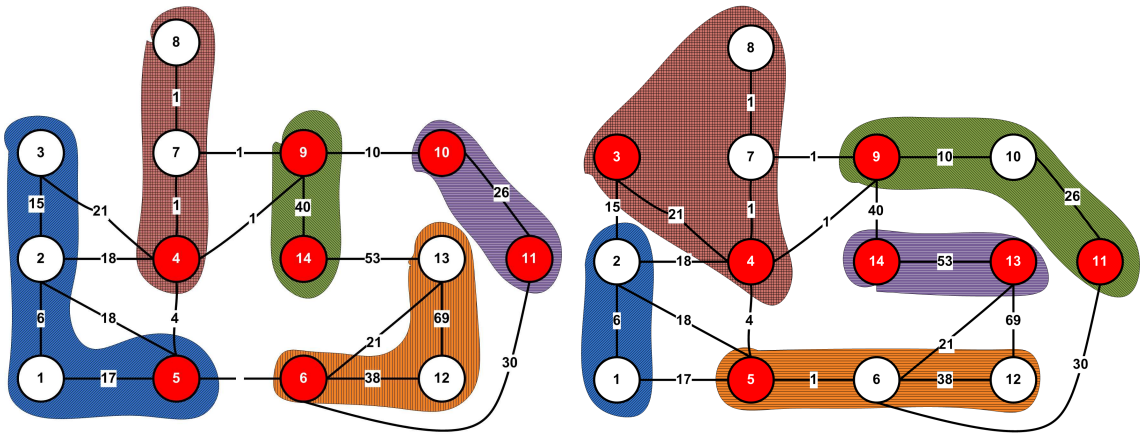
Table 4.31 shows the resulting delay and the thresholds that those paths were compared to.

Table 4.31: Message paths traversed for the 14 Node 3 message slowest case in Windows

	Nodes Traversed					Time	tN Delay	Total Delay	Threshold
Path 1	1	2	4			24	0	24	600
Path 2	2	5	6			57	600	657	1200
Path 3	8	7	4	5	6	7	1200	1207	1800

4.12 Windows VS Linux

As stated before, this research was done utilizing two versions of the XPress-MP optimizer; a Linux version and a Windows version. For unknown reasons, when running exactly the same scenario separately on each operating system; the result obtained was different. We speculate that this is a difference in random number seeds, which resulted in different search patterns of the solution space. An example of this discrepancy is shown in Fig.4.18. As we can see in this figure, the configuration resulting from each of the operating systems is completely different. The domains formed are different, the trust nodes are placed in different nodes, too. However, both of the solutions comply with the optimization constraints. This difference of results was not always the case, but it often presented itself throughout the research. Since the results were verified to be feasible answers, these differences were assumed to be correct.



(a) Windows Run (b) Linux Run

Figure 4.18: Fourteen Node scenario results from Windows and Linux

4.13 Summary

The trials described in this chapter give a clear view of what this optimization model does. The model produces reliable configurations regardless of platform utilized. In each trial shown the model partitioned the network into domains and placed trust nodes at locations where it was able to protect the integrity of all communications between domains. But most importantly, the model is able to produced an output with the above characteristics while respecting the constraints of response time and domain contiguity. In this chapter, we saw that the model produces different configurations based on the number of trust nodes available and the minimum number of nodes per domain. When the model was alloted a large number of trust nodes in comparison with the total nodes in the network, the model would produce as many domains as possible, which is what our objective function is.

V. Conclusions and Recommendations

5.1 Chapter Overview

This chapter gives a research overview, summarizes research findings, establishes the significance and impact of this research, provides conclusions, and the potential impact that the results of this thesis might have for future work. Additionally, and finally it makes recommendations for follow-on work in this area.

5.2 Research Overview

Most of our critical infrastructure and especially our electric utility industry operates under tight conditions. The demand for services has grown while the transmission system's capacity has experienced slower growth. This has caused the system to become more unstable and has increased the risk of failure. The SCADA system that monitors this delicate balance is modernizing and taking in new technologies that bring in newer capabilities but at the same time, new vulnerabilities. Consequently, security becomes paramount in an environment where our critical infrastructure are a target of attacks that could weaken our economy. Therefore, there is a strong need to protect our essential infrastructure with the utilization of new technologies. A careful design of the network topology and the implementation of a network security-based trust system added to the SCADA provide an extra layer of protection against large attacks that may render our society vulnerable. The challenge is to do so in a way or ways that will not disrupt the time-critical protection and control systems in a SCADA system The research in this article addresses this challenge through a trust system placement algorithm.

5.3 Summary of Research Findings

This research thesis explored the use of a new software program created specifically for this research. This software utilizes linear programming techniques to demonstrate that the fielding of the trust system along with the

compartmentalization of a SCADA or power grid network is possible, and more importantly; it is safe.

This research showed that the software, was accurate in producing feasible solutions within a reasonable length of time, usually seconds or maybe a few hours. Although, we can see that as the size of the network increased, the running times increased as well. However, the larger running time increase was mainly due to the values entered in the maximum number of trust nodes and the minimum number nodes per domain; as these determine the size the solution space that the optimizer will have to search. However, these times could considerably be reduced if the optimizer was to be processed in higher end computer architectures.

The software developed was shown to perform efficiently and accurately under different variations of the input scenario. The software was run against a total of 220 runs, each run represented a different network with either a different number of buses (nodes) or the nodes connected by a different set of edges (branches). We also varied the number of trust nodes that were available, the minimum number of nodes that could be assigned to a domain. Moreover, we modified the number of messages and their paths, to check for response time and check if they were violated when the configuration was produced.

Furthermore, we took a configuration produced by a prior trial, used the path delay in the output to create a threshold file with reduced values, such that those paths were violating these thresholds from the beginning. And we ran this scenario against the modified time threshold, and compared the results. We found that the optimizer produced a different network configuration that enforced the new time constraints. What this means is that the optimizer formed different domains and moved the location where trust nodes where assigned in order to meet the new constraints.

These findings are crucial, because they demonstrate that the proposed approach to the implementation of the trust system is not only possible, but safe.

5.4 Conclusion

This research demonstrated that the trust system proposed can be implemented in a real life network by adding trust nodes to strategic nodes and combining it with a methodical compartmentalization of the SCADA network that increase security of the network. It indicated that we can obtained the benefits that the trust system provides and the security enhancements that come with it by effectively determining where this trust nodes should be placed in the network.

While the application of this proposed approach was focused on the electric power grid. It can easily be applied to other industries in the critical infrastructure. And also, they can be implemented in environments where requirements are not as strict with similar results as well.

Finally, the proposed approach to trust system implementation appears to hold great promise to facilitate greater interconnected communication in the electric power grid. Additionally, this approach seeks to provide increased safety that can result through secure message sharing, facilitated by the trust system and domain grouping. This system is a step towards a comprehensive security architecture for the power grid.

5.5 Significance of Research

The sector of our economy that this research targets is an extremely critical one. It is one where although new communication paradigms and technologies are being introduced without a thorough understanding of the consequences. Nonetheless, the security issues are not being explored and solutions or alternatives are still in their early stages. We have to remember as a nation that we are under a new type of conflict and that our enemies are not government, or armies. We face a new type of warfare that is not directed to our military defenses but to our society, to our economy; and we need to protect these to the utmost. We cannot afford to ignore security as a discipline and daily practice. This research is of great importance

because it constitutes a move forward toward a secure and protected infrastructure, so critical to our country.

This research brings together previous thesis efforts that proposed security alternatives to improve the protection of the facility. This thesis implemented those ideas and tested them against data that represents real world systems. The fact that those ideas can be fielded and add critical security functions such as firewall, internal traffic protection, trust level implementation, message encryption/decryption, and other functions; without risking the safety of the facility is a step forward toward a more secure operation at a time where is urgently needed to better protect our critical infrastructure. Furthermore, these encouraging results represent are valuable because this approach can be fielded without requiring interruption or shut down of the system. It can be implemented almost transparent to operations.

5.6 Recommendations for future work

Although the results were very promising, there is further work to be done to make it a more solid alternative. One is that the optimization model processed could be processed utilizing a parallel processing and process more complex data sets, as the 57 nodes was not run in its entirety since both operating systems dumped the process when the operating system resources became scarce. Also, there is the option of upgrading the random access memory installed in the Linux computer, since it is a 64 bit architecture. This could potentially allow completing processing the larger runs.

Additionally, the model could be modified and implemented as a quadratic programming model. This may be useful as some of the constraints such as restricting the range of nodes added to a domain so that the the domain node count is close and the creation of large domains and very small domains is not allowed.

Also, the configurations that resulted from the experimentation in this research may be tested in network simulators. Network simulators could be used to validate

and support the results obtained by this research, by emulating and graphically displaying the behavior of the network with a load of messages.

VI. Appendix 1

The figure shown in this section, presents the configuration produced after six hours of processing a 57 node network. This configuration was initially ran without a time limitation. However, the process was halted after three days and no output collected. The figure present several domains that have been formed but most of them violate the constraint of domain contiguity we can see portions of one domain in different parts of the network. This situation might have occurred because the optimizer was not allowed to finish running, but instead forced its termination by placing a time limit. Next page, will show its output.

Blank space left intentionally

Figure A.1: 30 Node network, Maximum of 10 trust nodes, and a minimum of 6 nodes per domain

Bibliography

1. *Xpress-Mosel Language Refernce Manual*. Dash Optimization, release 1.2 edition, June 2003.
2. "The new York City Blackout". World Wide Web, Oct 2008. URL `http://www.imap.net/who_we_are/crothers_2003.php`. Accessed on Oct 08.
3. Boyer, Stuart A. *SCADA Supervisory Control and Data Acquisition*. ISA, 3rd edition, 2004. ISBN 1-55617-877-8.
4. Cardenas, Alvaro A. "Research Challenges for the Security of Control Systems". Under submission.
5. Coates, Gregory M. *Collaborative, Trust-Based Security Mechanisms for a National Intranet*. Master's thesis, Air For ce Institute of Technology, 2005.
6. Dagle. "Supervisory Control and Data Acquisition (SCADA) Introduction", September 2005.
7. Eaddie, M. *Dialable Cryptography for Wireless Networks*. Master's thesis, Air Force Institute of Technology, 2008.
8. Erwin, Michael C. *Combining Quality of Service and Topology Control in Directional Hybrid Wireless Networks*. Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, June 2007.
9. Grimes, Mark. "SCADA Exposed". MS Power Point Presentation for SAIC.
10. Hopkinson, Kenneth PhD. "EPOCHS: A Platform for Agent-Based Electric Power and Communication Simulation Built From Commercial Off-the-Shelf Components". *IEEE*, 21(2):548 – 558, May 2006.
11. Hopkinson, Phil. "H Volt Power Transformer Consultant Interview". Interview, November 2008. URL `http://www.hvolt.com/hopkinson.htm`. Telephone Interview.
12. http://en.wikipedia.org/wiki/Chernobyldisaster. "Chernobyl Disaster on Wikipedia". internet, May 2008. Accessed on May 15th 2008.
13. http://en.wikipedia.org/wiki/Oil_refinery. "SCADA". Accessed 10 May 2008.
14. http://en.wikipedia.org/wiki/SCADA. "SCADA". Accessed 10 May 2008.
15. Krutz, Ronald L. PhD. *Securing SCADA Systems*. Wiley Publishing, Inc., 2006. ISBN 0-7645-9787-6.
16. Labs, Sandia. "http://www.sandia.gov/scada/faq.htm". Internet. Accessed 20 May 2008.

17. Lewis, Ted G. *Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation.* John Wiley and Sons, Inc., 2006.

18. Marek, Zima. "Design Aspects for Wide-Ara Monitoring and Control Systems". *IEEE*, 93, May 2005.

19. Morgan-Orth-Repik. *Supervisory Control and Data Acquisition (SCADA) Threats, Vulnerabilities and Forensics.* Final project, Air Force Insitute of Technology, August 2007.

20. Moteff, Paul, John / Parfomak. "Critical Infrastructure and Key Assets: Definition and Identification". PDF Report, October 2004.

21. Njemanze, Hugh. "SCADA Security Protections are on the Increase". *Reprinted from Pipeline and Gas Journal*, 2007.

22. of North Texas, University. "The Portal to Texas History". website, September 2008. URL `http://texashistory.unt.edu`.

23. http://world nuclear.org. "World Nuclear Associaion", April 2009. Accessed on April 2009.

24. Orlin, Ahuja/ Magnanti /. *Network Flows: Theory, Algorithms, and Applications.* New Jersey: Prentice Hall, 1993.

25. OutageTaskForce. *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations.* Technical report, U.S.-Canada Power System Outage Task Force, April 2004.

26. Peterson, Dale (editor). *Proceedings of the SCADA Security Scientific Symposium.* Digital Bond Press, 2007.

27. Pollet, Jonathan. "Top 5 Security Issues with Securing Real-Time Control and SCADA Systems". PDF/Power Point Presentation, June 2005. SCADA/DSC Security Engineer.

28. Roberts, Gregory R. Capt. USAF. *Evaluating Security And Quality Of Service Considerations in Critical Infrastructure Communication Networks.* Master's thesis, Air Force Institute of Technology, 2008.

29. Shaxia, Vittal, Xie /G. Manimaran/ Vijay. "An information Architecture for Future Power Systems and its Reliability Analysis." *IEEE*, 17(3):837–863, August 2002.

30. Varnado. "SCADA and the Terrorist Threat: Protecting the Nation's Critical Control Systems Statement of Dr. Samuel G. Varnado", October 2005.

31. University of Washington, College of Engineering. "Power Systems Test Case Archive". World Wide Web, Aug 2003. URL `http://www.ee.washington.edu/research/pstca/`. Accessed on August 2008.

32. Wikipedia. “Electric Power Transmission”. World Wide Web, Sep 2008. URL http://en.wikipedia.org/wiki/Electric_power_transmission. Accessed on September 2008.

33. Wikipedia. “Great Storm of 1987”. World Wide Web, Nov 2008. URL http://en.wikipedia.org/wiki/Great_Storm_of_1987. Accessed on October 2009.

34. Wikipedia. “March 1989 Geomagnetic Storm”. World Wide Web, Oct 2008. URL http://en.wikipedia.org/wiki/March_1989_geomagnetic_storm. Accessed on Oct of 2008.

35. Wikipedia. “Northeast Blackout of 1965”. World Wide Web, Jan 2009. URL http://en.wikipedia.org/wiki/Northeast_Blackout_of_1965.

36. Winston, Wayne L. *Operations Research Applications and Algorithms*. Curt Hinrichs, fourth edition edition, 2004. 2004.

37. Wood, Allen J. *Power Generation Operation and Control*. Wiley-Interscience Publication, 2nd edition edition, 1996. ISBN 0-471-58699-4.

38. Zima, Marek. *Special Protection Schemes in Electric Power Systems*. Literature survey, Swiss Federal Institute of Technology Zurich, June 2002. Eeh Power Systems Laboratory.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
18–06–2009	Master's Thesis	June 2007—June 2009

4. TITLE AND SUBTITLE

An Efficient and Effective Implementation
the Trust System
For Power Grid Compartmentalization

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Juan M. Carlos Gonzalez, Capt, USAF

5d. PROJECT NUMBER

09-267

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GCS/ENG/09-01

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIGA (John Matyjas)
525 Electronics Parkway
Rome, NY 13441
DSN 587-4255, e-mail: john.matyjas@rl.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RIGA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The goal of this research is to show in a simulated environment that security of the network can be strengthen by first fielding the trust system and second, by dividing a network into smaller clusters, called "domains", in order to isolate anomalies or intrusions detected. In order to show this, a mathematical model of the problem will be built and translated into a software tool that at the end will receive real-life-network data as input. This program uses real world power grid representative data, outputs a network configuration that has used the concepts described above of network compartmentalization and strategic placing of trust nodes. As a result, this new network configuration ensures safe day-to-day operations by minimizing the effects in case of an attack or equipment malfunction of the system by subdividing the network into domains. Each domain protected by a trust node(s) without violating timing constraints.

15. SUBJECT TERMS

SCADA; network security; trust system; power grid; critical infrastructure; three dimensional.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Kenneth M. Hopkinson, PhD
U	U	U	UU	121	19b. TELEPHONE NUMBER *(include area code)* (937)255–3636, ext4579; kenneth.hopkinson@afit.edu

Standard Form 298 (Rev. 8–98)
Prescribed by ANSI Std. Z39.18