12-18-2011

Computational Modeling for Transportation Network Security

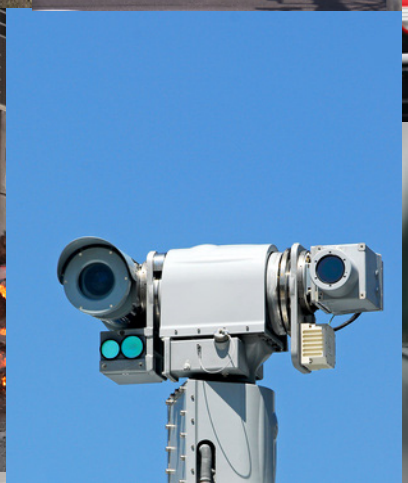
Sherif A. Tolba
University of Connecticut, sherif.tolba@engr.uconn.edu


BRIDGE ROAD
BURWALLS ROAD
Options


50
RADAR


SPEED LIMIT 45


NO TURN ON RED 6AM-9AM MON-FRI
CAUTION
Park
SPEED CHECKED BY RADAR
NO SKATEBOARDING OR ROLLERBLADING
SPEED LIMIT 25
WRONG WAY
ONE WAY
STREET SWEEPING
3rd MONDAY EVERY MONTH 7AM - 10AM




40

Computational Modeling for Transportation Network Security

Sherif Ahmed Tolba

B.Sc., Mansoura University, 2007

A Thesis

Submitted in Partial Fulfillment of the

Requirements of the Degree of

Master of Science

at the

University of Connecticut

2011

APPROVAL PAGE

Master of Science Thesis

Computational Modeling for Transportation Network Security

Presented by

Sherif Ahmed Tolba, B.Sc.

Major Advisor _____
Reda A. Ammar

Associate Advisor _____
Sanguthevar Rajasekaran

Associate Advisor _____
Nicholas Lownes

University of Connecticut

2011

Computational Modeling for Transportation Network Security

ABSTRACT

Computational Modeling for Transportation Network Security. (December 2011)

Sherif Ahmed Tolba, B.Sc., Mansoura University

Chair of Advisory Committee: Dr. Reda Anwar Ammar

The transportation system is among the different systems that require high levels of security. Threats that endanger security can be any external factors which may cause the system to malfunction or even fail. This work addresses two important elements having a significant effect on the security of the transportation network. The first is the dynamic interaction between an attacker targeting the network and the entity which protects it. The second is the technology used to protect the network and the assurance of its suitability and proper functionality.

Modeling the interaction between the attacker and the defender is important for understanding the evolution of each side and being able to make informed decisions about the ways of defense and technology deployment. There have been several attempts to model this interaction, but including the perceptions of both participants in the model is very rare. One contribution of the proposed model is the inclusion of these perceptions and exploiting them to support technology deployment decisions. The system dynamics modeling approach is used to build the interaction model. The model takes a set of intelligence-based input variables and produces a temporal road-map for technology deployment plans taking into consideration link (road) ranks and many other important parameters such as external real-world factors.

Results show that technology deployment does not necessarily need to be implemented all at once; instead, it can be spread over an appropriate time period. They also show that the protection agency's uncertainties have big impact on the outcomes of this interaction. Finally, it was also concluded that a suitable representation of what represents a chance for the attacker is very important for producing accurate model outputs and making correct decisions.

In addition to having a well studied temporal road-map for technology deployment, it is necessary to select a suitable deployment and assure the proper functionality of the deployed technology. This is what the second part of the work is concerned with. There are several technologies that can help securing the transportation network. Attention is given to Wireless Sensor Networks (WSNs) as they offer countless security applications. Two deployment algorithms, which can be applied to transportation networks of different scales, were developed along with a software tool that facilitates the use of these algorithms by protection agencies.

DEDICATION

This work is dedicated to my Lord, my parents, my wife, my beloved son, and the whole world. Without the help of my Lord and that of each one of the above persons, I wouldn't have been able to finish this thesis.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Reda Ammar, and my committee members, Dr. Sanguthevar Rajasekaran, and Dr. Nicholas Lownes, for their guidance and support throughout the course of this research.

Thanks also go to my friends and colleagues, especially Lance, and the department faculty and staff for making my time at UConn a great experience. I also want to extend my gratitude to the Department of Homeland Security and the National Transportation Security Center of Excellence at UConn, which provided the funding and support.

Finally, thanks to my parents for their strong support and encouragement, and to my wife for her patience, support, and love.

NOMENCLATURE

ANPR	Automatic Number Plate Recognition
ARA	Adversarial Risk Analysis
BS	Base Station
CAS	Collision Avoidance System
CBRNE	Chemical, Biological, Radiological, Nuclear, and Explosive
CC	Command Center
CCTV	Closed Circuit Television
CG	Computational Geometry
DRT	Deployment Realization Tool
GIS	Geographic Information Systems
GN	Gateway Node
GPS	Global Positioning System
HS	Homeland Security
ITS	Intelligent Transportation Systems
MAT	Medial Axis Transform
MS	Mobile Sink
P^2I^3	Perception squared Interaction cubed model
PGIS	Parking Guidance and Information System
QoS	Quality of Service
RFID	Radio Frequency Identification
RSU	Road Side Unit

SN	Sensor Node
TMC	Traffic Management Center
V(D)MS	Variable (Dynamic) Message Signs
V2V	Vehicle to Vehicle (Communication)
VII	Vehicle-Infrastructure Integration
VIP	Video Image Processors
VLSI	Very Large Scale Integration
VS	Vice Sink
WSN	Wireless Sensor Network

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

Page

Page

Page

Page

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION

1.1. Introduction

Transportation network security is of utmost importance. Similar to the other vital infrastructures such as power plants, water and sewer systems, oil plants, and military bases, the transportation network has a significant impact on the flow of daily life tasks. Economy is directly affected by the state of transportation networks. Public welfare and prosperity are other elements that get affected greatly by the state of the transportation network. Due to the recent and continuing attacks and disasters that strike different transportation network elements, it has become very important to provide suitable and efficient means to secure the transportation network. This, of course, includes all modes of transportation and all supportive structures and services. Homeland security agencies and governments have become interested more and more in utilizing well established and emerging technologies and techniques to achieve the desired level of safety. Large sums of public and private money have been invested in developing and inventing new methodologies to achieve and maintain these security levels. The targeted technologies have a wide range, starting form very simple screening processes to advanced technological systems intended to acquire and analyze data and make sophisticated decisions.

In this work, the main focus is on developing techniques supported by software tools to aid technological planning and decision support for homeland security agencies. The work is divided into two basic parts; one is associated with provision of a technology

deployment road-map, and the other is associated with the development of a WSN deployment algorithm to aid the technology deployment process.

First, a model and an accompanying tool are developed to facilitate the planning process of technology deployment in transportation networks. The developed model takes a set of intelligence data and produces a road-map for sensor deployment in terms of the time frame for conducting the deployment and the number and the selection of the links that will be protected.

Second, close attention is given to the deployment process, taking into consideration the needs of the routing protocol which will be used in the deployed network. Two deployment algorithms were developed. Also, a software tool aiming at serving HS personnel in making use of these algorithms to obtain deployment information (such as sensor densities and distribution) was developed.

Discussion of the results obtained using the developed model and algorithms are provided. A discussion of the usefulness and practicability of the results is also included.

1.2. Problem Summary

This section provides a quick overview of the addressed problems and their breakdown. The attacker-defender interaction model is first discussed. Then a summary of the deployment problem is given.

1.2.1. Attacker-Defender Interaction Model

It is doubtless that the mental power of human beings is superior to all man-made technologies. This has been the reason for continuously having different successful attack scenarios against transportation networks at different places in the world. Whenever a

homeland security agency uses a specific technology or approach, the opponent tries to find vulnerabilities in that technology and utilizes them for his advantage. The natural consequence is that the HS agency then tries to develop new ways and technologies that make it more difficult for the attacker to succeed. Then the scenario repeats again and again. Hence, it is noticed that there is an interaction between the two opponents where there are turns of success and failure.

Considering the perceptions of both sides of the interaction is important for HS agencies in order to enable the prediction of future events and try to stop them or mitigate their effect if they happen. The interaction is modeled in this work to study the evolution of both parties over time and to provide a road-map (time plan) for sensor deployment along with the number and the selection of transportation network links to be protected at each point in time.

1.2.2. WSN Deployment and Packet Routing

The model that was briefly discussed in the previous sub-section had the target of providing the plan for deployment and the time frame for doing so. After having this information, the HS agency is supposed to select and deploy an appropriate technology to achieve the desired protection goal. Wireless sensor technology is has recently been one of the candidates due to its diverse military, civilian, and environmental applications. We select WSNs technology for use in this work and specifically for the purpose of detecting Chemical, Biological, Radiological, Nuclear, and Enhanced Explosive (CBRNE) devices. Currently available wireless sensors used for this purpose are expensive and bulky. Research is being conducted to reduce the prices and sizes of such sensors while maximizing the detection capabilities. Our goal is to provide a deployment plan to allow

protecting network links while achieving the highest performance for the deployed network. We believe that achieving high performance in terms of energy consumption, end-to-end delay, throughput, network lifetime, reliability, etc. cannot be achieved effectively if WSN deployment and packet routing protocol design are separated. Therefore, we consider many routing-related parameters such has hole avoidance and energy depletion while designing the deployment algorithm. On the other hand, routing protocols should be designed in such a way that allows the best use of the deployment features. The contribution of this part is the proposal of two deployment algorithms that take into consideration link (road) importance from the transportation perspective and protocol performance metrics from the WSN perspective.

1.3. Security: Basic Definitions

Before delving into the details of the interaction between attackers and HS agencies and the deployment of protective technology to secure transportation networks, it is important to understand the meanings of different security-related terms. This will make the following discussions easier.

1.3.1. Hazards and Threats

Hazards and threats are very closely related. They can be simply thought of as a cause and a result. Hazard is the cause and threat is the result. Green (2008) and SRA[1] defined hazard as "a condition or a physical situation that has the potential to cause harm". For example, a hurricane or an earth quake represent a hazard. A possible attack to some critical infrasture is also considered as a hazard. Willis et al. (2005) defined a threat as "the probability that a specific target is attacked in a specific way during a specified time

[1] The Society for Risk Analysis : http://www.sra.org/resources_glossary_g-i.php

period*"*. This simply means the probability that an attack occurs. Of course this is in the context of human caused hazards and the resulting threats. A similar definition is also applicable for natural or technology caused (such as a nuclear plant failure) hazards. A threat was defined by Green as "the expected impact of a developing hazard, and the probability that this impact will work against our vulnerabilities". From the above definitions, we can provide the following definitions for hazards and threats:

Hazard**:** *An expected or probable dangerous event whether natural or man-caused.*

Threat**:** *The probability of occurrence of a hazardous event and the negative impact that it causes.*

We also note that hazards lead to threats or pose threates of different levels.

1.3.2. Impact / Consequences / Damage:

These three terms are usually used interchangeably to express the result of a hazard event that has been realized. The term 'damage' is straight forward. Impacts and consequences are used to express damage because the impact or the consequence of an event that is involved in risk analysis usually, if not always, causes some kind of damage to the system. However, a distinction should be made between the latter two terms as pointed out by Green: impacts are considered "a result of the event" while consequences represent the "result of the interaction of the impact with other systems". In addition, he distinguished them in terms of their time span; impacts are short term while consequences are long term. The following definitions for the above three terms can be concluded:

Damage: *Physical (or sometimes incorporeal) undesired results of a realized hazard*

Impact: *Short term, event-induced damage*[1]

Consequences: *Long-term results of the impact's influence on other systems*[1]

1.3.3. Vulnerability

There are several definitions for vulnerability that have been found. For example, Willis et al. defined vulnerability as the conditional probability of an attack causing damage given *"a specific attack type, at a specific time, on a given target"*. Another definition given by Wetsen (2005) is: *"the degree of loss caused to an element or a set of elements due to the occurrence of a hazard of a certain magnitude"*

The above definitions and all other existing ones agree on the fact that vulnerability deals with the exposure of a system to the hazards and/or attacks. In other words, as stated by Green, vulnerability reflects *"how well or bad a system is protected"* or is secure in the face of threatening hazards. Below, we provide our definition of vulnerability.

Vulnerability: *A measure of the degree of damage that could be encountered (due to the system's exposure and susecptability to being damaged) if a hazard is realized*

1.3.4. Risk

The definition of risk may differ slightly from one context to another (Wetsen). The classic risk model defines risk as the product of the probability of event (e.g. attack) occurrence and the damage of that event (Amenaza (2003), Rausand (2005), and Green). Another very common definition of risk that is, in fact, a further breakdown of the previous one was provided by Ezell et al. (2010), Willis et al., and Amenaza. That

[1] Used the same distinction that Green (2008) provided

definition expresses risk as the product of three terms; the probability that an attack or a hazardous event occurs "threat", the probability the attack succeeds, i.e. causes damage, given that it occurs "Vulnerability", and the consequences of the event "consequence". This indicates that the probability of occurrence of the attack event is in fact the product of threat and vulnerability as pointed out by Amenaza. In addition to the mathematical product of these factors, Willis et al. provided the following definition for risk which we adopt as well:

Risk: *"The expected consequence of an existent threat for a given target, attack mode, and damage type"*

We adopt similar mathematical definitions for risk, threat, consequences, and the attack-event probability to the ones provided by Willis et al., Amenaza, and Ezell et al. as follows:

$$\text{Risk} = \text{Event Probability} \times \text{Consequence} \quad (1)$$

$$\text{Event Probability} = \text{Threat} \times \text{Vulnerability} \quad (2)$$

$$\text{i.e.} \quad \text{Risk} = \text{Threat} \times \text{Vulnerability} \times \text{Consequence} \quad (3)$$

1.3.5. Risk Analysis

As was noticed from the above definition of risk, it is concerned with threats, vulnerabilities, and consequences. Risk analysis as a result is the process of identifying hazards, their associated threats, system vulnerabilities to the threats, the consequences of their realization, analyzing these threats and vulnerabilities, estimating the risk, and trying to reduce or eliminate the consequences using appropriate countermeasures. There are several standards for risk, risk management, and risk assessment/analysis that are

based on the context in which it is dealt with. For example, ISO 17799[1] is an information security standard that includes many sections; one of them is associated with "risk assessment and treatment". According to that standard, risk analysis is intended to *"identify threats and vulnerabilities and analyze them to determine the exposure of a system and try to reduce or eliminate impacts"*. Another standard is NORSOK[2]; for risk and emergency preparedness analysis, which describes risk analysis as the process of *"using available information to identify hazards and estimate risk"*.

In general, risk analysis can be classified into two categories in terms of the analysis approach: qualitative and quantitative risk analysis. Qualitative risk analysis is simply based on rating. The risk is rated on a scale of high, medium, and low. This approach is common due to its simplicity as it does not require knowledge of the probability of event occurrence. This approach provides as output the risk level. Quantitative risk analysis, on the other hand, requires knowledge of the event probability and the consequences. The quantification of this probability is usually the problem (Amenaza, 2003) and is case specific. It is also difficult to accurately expect and quantify the consequences. At the end of this approach a value for the risk based on the product of event probability and the consequences is provided.

1.4. Security in Transportation Networks

The diversity of transportation modes and the complexity of the associated infrastructure make their protection a manifold process. Taking the road network as an example, there are several aspects related to securing them, such as securing highways,

[1] ISO 17799: http://17799.denialinfo.com/risk.htm
[2] NORSOK STANDARD Z-013, "Risk and emergency preparedness analysis", Norwegian Technology Centre, Rev. 2, 2001

city streets, bridges, tunnels, bus systems, traffic control centers, city entrances and exists, traffic control software systems, etc. Another example is the railroad systems where security achievement requires the protection of the stations, vehicles, tracks, and other related operational software systems. The process of securing the above and other transportation modes includes the deployment of protective technologies for the infrastructure and the use of efficient strategies and practices. Traditional means of security in transportation include video surveillance via Closed Circuit Televisions (CCTVs), passenger and baggage screening at access points, and intelligence information. The power of these techniques has decreased over time as new threats began to arise due to technological advances. This required the development and exploitation of more advanced technologies to cope with such threats. The development of intelligent transportation systems (ITS) with the aim of achieving comfort and safety to passengers as well as maintaining the smoothness of traffic, maximizing time savings and economic progress, and enhancing the quality of life, is another booster of the technological advances that support transportation security. With the advance of this field, several technologies that were already there, and new ones that emerged as a result of the dire needs of that field, became of significant importance to serve both intelligent transportation objectives and security requirements. In the next section, we give an introduction about intelligent transportation systems and discuss some of these technologies.

1.5. Intelligent Transportation Systems and Security Improvement

Intelligent Transportation Systems abbreviated as ITS, represent transportation systems that exploit information and communication technologies. The use of these

systems has numerous objectives, including: 1) passenger convenience; in terms of travel time, congestion reduction, and accident avoidance, 2) traffic management; by controlling traffic flow, collecting traffic data and extracting traffic patterns and other important traffic information, and giving travel guidance information to passengers, 3) emergency response; through real-time detection of accidents, retrieval of accident information, notification of other passengers, and the notification of emergency crews about the accident's location and severity, 4) low enforcement; by detecting speed and other regulation violations, 5) security guarantee; through the detection of suspicious behaviors, carriage of explosive devices, and other technological or natural hazards, 6) economic benefits; by smoothing the flow of businesses and reducing fuel consumption.

Numerous technologies and applications for addressing the above needs exist and some have been already implemented in different countries like The United States, United Kingdom, Japan, Singapore, and others. Other technologies are being developed with the aim of using them to improve transportation network security. Examples of existing technologies include: 1) Automatic Number Plate Recognition (ANPR) - application; used in several contexts such as parking, access control, road tolling, border control, trip time measurement, and low enforcement[1], 2) vehicle detection and surveillance technologies, Mimbela et al. (2007) - technology; classified as intrusive and non intrusive sensor technologies. Intrusive sensors require some sort of embedment into the roadway or its sub-layers. They can be fixed on top of the road surface, inside the pavement, or in the subgrade of the road. Non-intrusive sensors need not be embedded in the road. They can be placed either on the road or alongside the road. Examples of intrusive sensors include inductive loops, weigh-in-motion sensors, magnetometers,

[1] http://www.anpr.net/anpr_09/anpr_applicationareas.html

piezoelectric cables, and microloops. Non-intrusive sensors include, video image processors (VIP), microwave radar sensors, ultrasonic, passive-infrared, laser radar sensors, and passive acoustic sensors, 3) Closed Circuit TVs (CCTVs) - technology; closed circuit means that the captured video is sent to a specific place, usually a control room located at the Traffic Management Center (TMC), as opposed to being broadcast and being publicly available. It is used to provide a means of monitoring roadways for the purpose of incident quick response, incident severity determination, and travel advisory services, FDOT (2011), 4) Variable (Dynamic) Message Signs (VMS or DMS) - application; electronic message boards used to inform travelers of congestions, road blockages due road work, accidents, or other incidents (FDOT), 5) Global Positioning System (GPS) – technology/application; there is no doubt that GPS is a ubiquitous technology and is one of the technologies that inherent to the intelligent transportation system. As is well known, GPS's guide passengers to their destinations, provide arrival time information, and enable passengers to select either shortest path or shortest time routes. They also have the ability to detour passengers upon request when there is a traffic jam, 6) Dynamic Traffic Light Sequence - application, Al-Khateeb (2008); traditionally uses inductive loops, video image processing, or beam crossing to detect vehicles and then schedules the traffic light time sequence in real-time. Recently, with the development of Radio Frequency Identification (RFID), many suggestions to use this efficient technology in vehicle detection and light sequencing were made. The brief idea is that RFID readers are mounted at the intersections, while RFID tags are attached to different vehicles. The readers detect different vehicles' arrival times and locations then start sequencing the traffic light in real-time to allow smooth traffic flow, 7) Variable

Speed Limits[1] - application; usually used to reduce traffic speed in bad traffic or weather conditions. In the past, they were controlled manually from the traffic management center based on the observed road or weather conditions, and recently they have been automated. Many countries already have them in place such as the United States, Germany, Britain, New Zealand, and Austria. They are usually placed on the road segments with extremely varying road conditions, dangerous road topology like steep roads or roads with hidden bents, or where many accidents frequently happen.

The above are only few examples of some ITS technologies and applications. There are many others such as Collision Avoidance Systems (CAS), emergency vehicle notification systems, automatic traffic enforcement, smart traffic evacuation management systems, Vehicle-Infrastructure-Integration (VII), Vehicle-to-Vehicle communication (V2V), etc.

1.6. Modeling Approaches Used in Transportation Security

In the previous section, we listed some of the existing and emerging ITS technologies and applications. As stated previously, two of the goals of these technologies and applications are security guarantee and assistance of emergency response. Another very important assistive approach in this regard is modeling and simulation. Through modeling and simulation, informed decisions can be made which should, together with these technologies, lead to the sought results. Several approaches for modeling the interactions between attackers and HS agencies, analyzing the risks of attacking the system, and quantifying these risks exist in the literature (See section 3.1 for more details). In this section, we give a brief introduction about some of these approaches.

[1] http://en.wikipedia.org/wiki/Speed_limit#Variable_speed_limits

1.6.1. Descriptive vs. Prescriptive Models

It is important recall the difference between to types of mathematical models before starting the modeling process, namely descriptive and prescriptive models. This should make the model design process structured, clear, and avoids confusing the objectives of different parts of the big model. Descriptive models, as their name implies, describe and simulate the behavior of a system or a device. On the contrary, prescriptive models try to find the best result for some problem. This result can then be used to prescribe a course of action to be followed (S. Chapra and R. Canale, 2006).

In this work, we use a mixture of descriptive and prescriptive models as will be seen in Chapters 2 and 4. The model we propose consists of three sub-models. Two of them are descriptive (RW and Attacker Perception models) while the third (HS Perception model) is a prescriptive model. Furthermore, the big model that consists of these three sub-models can be classified as prescriptive, because the main goal is to provide HS personnel with a decision support system.

1.6.2. Game Theoretic Approaches

An approach used to study and analyze strategy interactions between two or more intelligent players. Each has a set of strategies to select from and achieves a certain payoff by choosing a specific strategy. Some of the common categories of games are: cooperative and non-cooperative, symmetric and asymmetric, zero-sum and non-zero sum, simultaneous and sequential, and perfect and imperfect information games. Non-cooperative games describe the details of all the available moves/strategies of the players, while cooperative ones abstract the game and deal with its big picture. Cooperatives

games usually allow the formation of 'coalitions' of players. In symmetric games, identities of the players can be changed without changing the payoff of the strategies[1]. Constant-sum games are games where the payoffs of the players add to the same constant sum. A special case of constant-sum games are the zero-sum games, where the gain achieved by one player corresponds to a loss by the other, and therefore, the total sum of the payoffs is zero (Cummings et al., 2006). In simultaneous games, the players move simultaneously, while in sequential ones, they move in alternating steps. Games also assume 'perfect/complete' or 'imperfect/incomplete' information, which describes whether each of the players knows the choices/actions of other players or not.

Strategies of the players can be either 'pure' or 'mixed' strategies. A pure strategy is the one that describes how a player responds to each game situation, while a mixed strategy is a probability distribution on the set of pure strategies; i.e. it picks a specific response with a specific probability. There are four main representations of a game, namely: the extensive form, the normal form, the characteristic function form, and the partition function form.

An important definition in Game Theory is the Nash equilibrium. Nash equilibrium is obtained when none of the players can achieve a better payoff by changing strategies. Game theory has applications in economics, political science, psychology, biology, computer science, and recently, in risk analysis. The literature contains many examples where Game Theory is used to model the interaction between defenders and attackers for the purpose of assessing system vulnerabilities and quantifying risk. Some of these examples will be discussed in chapter 3.

[1] http://en.wikipedia.org/wiki/Game_theory

1.6.3. Bayesian Decision Analysis Approach

An approach used for risk assessment in different fields such as medical and computer systems, intelligence analysis, and transportation security. It uses Bayes' theorem to find probability distributions of the system's random variables of interest based on the distributions of other random variables in the system. A network called Bayesian Network, represented as a directed acyclic graph, is used, where the nodes represent the random variables and the arcs represent the dependence of one variable on others. An associated algorithm is used to solve the network using Bayes' theorem and to find the probability distribution of the variable of interest in terms of the conditional and unconditional probabilities of the other random variables. The results are then used to update the network. As mentioned above, this approach is suitable for analyzing intelligence information and in making decisions. This is due to its ability to model complex interactions between variables, and dynamically update the probabilities (Cummings et al.).

Game Theory and Bayesian Decision Analysis are only two of the several modeling techniques that are used for risk analysis. For a more detailed discussion of the above and other techniques, we refer the reader to Ref. Cummings et al. (2006). In the next sub-section, we discuss system dynamics, the approach we used in this work.

1.6.4. System Dynamics Approach

System dynamics is an approach used to model complex systems and the interactions between their constituent variables. It was originally created to help corporate managers better understand their industrial processes (Radzicki and Taylor, 1997). Now, it is

widely used in different fields such as project management (Sterman, 1992), health applications (Homer and Hirsch, 2006), and marketing (Richardson and Otto, 2008). The two main components of system dynamics are causal loops and stock and flow diagrams. Causal loops are feedback loops that connect causes and effects in the system. Stock and flow diagrams are composed of two components: First, stocks, which are also called level variables, are variables that accumulate over time. This can be envisioned as the water level in a tank. Stocks do not disappear if time is hypothetically stopped (Kirkwood, 2010). Second, flows (rates), are exactly as their name indicates. They are rates of change of level variables. An example would be the control of the water valve used to fill the tank. Flows disappear if time is hypothetically stopped. An example system illustrating different system dynamics' concepts is shown in Fig. 1.

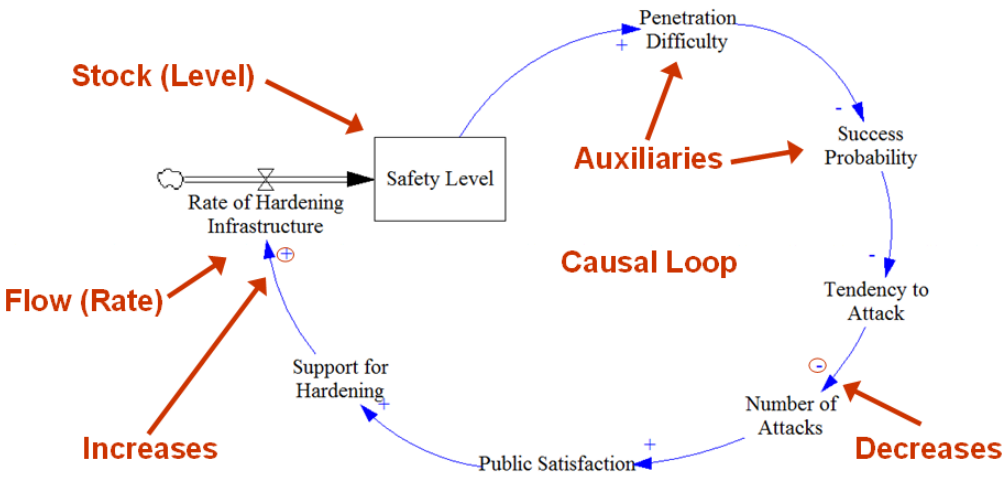


Figure1: An example system showing different system dynamics' concepts

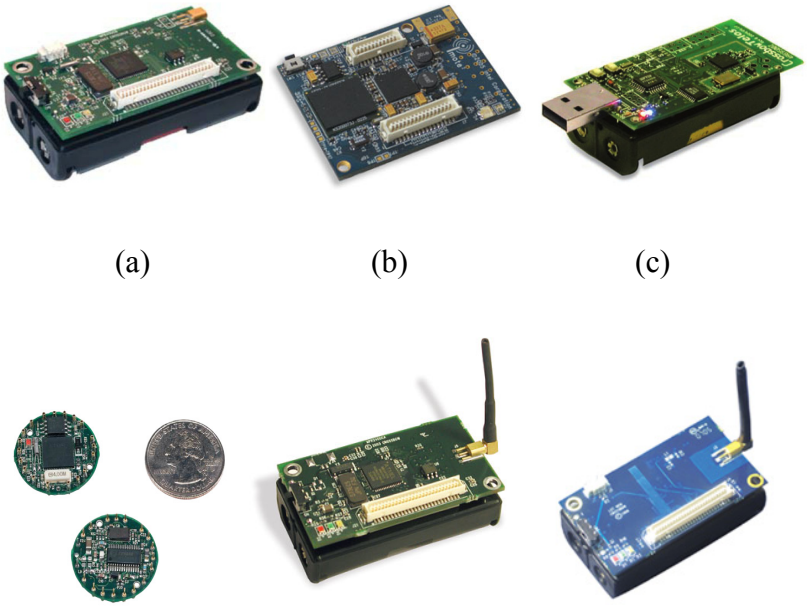
1.7. Wireless Sensor Networks

This section is organized as follows: first, we give a brief introduction about wireless sensor networks, then, the use of WSNs in transportation networks in discussed. Finally, we discuss both the deployment issue as well as data routing in such networks.

1.7.1. Introduction

Advances in the Very Large Scale Integration (VLSI) technology, wireless communications, Information Technologies (IT), sensing technology, and in the computational capabilities, has led to the development of Wireless Sensor Networks (WSNs). Attention of several the researchers has shifted towards the design and development of all technological aspects related to WSNs in the last two decades. The reason for this significant interest is the uncountable number of applications that they can be used in. Applications vary from military to civilian, industrial, and environmental applications. Examples of these applications include, but are not limited to: habitat monitoring, utility meter reading, border protection, structure conditions' monitoring, bomb detection, battle field monitoring, target tracking, health data collection, and many others. Another two very important applications for our purposes are: traffic data collection in ITSs, and transportation network security, through their use as a means of surveillance.

WSNs consist of small wireless devices called 'nodes' or 'motes' that have sensing, computational, and communication capabilities. They can sense temperature, humidity, acoustic signals, magnetic fields, light, direction of movement, speed, etc. However, they are limited in energy as they receive their power from a small battery. This feature is inherent to the design of sensor nodes; they are designed to be cheap and to have



(a) (b) (c)

(d) (e) (f)

Figure 2: Some popular WSN nodes: CrossBow's (a) MICA2 (b) IMote2 (c) TelosB (d) MICA2DOT (e) MICAz (f) IRIS

unattended operation. Usually, WSNs are deployed in large amounts unless is otherwise necessitated by the application. Some examples of popular nodes are shown in Fig. 2.

Because each node has limited energy, communication, and computational resources, a proper design of the network in terms of the deployment and the routing protocol is necessary. For this reason, a huge amount of literature exists in this field, trying to address different problems, in spite of the recent emergence of WSNs.

1.7.2. Use of WSNs in Transportation Networks

As in many other applications, WSNs find several uses in transportation networks. From a surveillance perspective, WSNs can be used in vehicle detection, classification, and reidentification (Cheung et al., 2007). Vehicle detection is done by means of either

the acoustic or the magnetic signatures which can be acquired by the sensors. The classification and reidentification of vehicles can also be done through magnetic signature analysis. Classification can also be done by the use of a vision-based sensor system.

WSNs can also be used for law enforcement through the use of visual sensors for automatic number plate recognition. Other applications of WSNs in ITSs, which assist in traffic management and passenger convenience, include traffic signal control (traffic light sequencing described previously), on-ramp metering, Parking Guidance and Information Systems (PGIS's), work zone management, road condition sensing, vehicle infrastructure integration, and vehicle-to-vehicle communication. Of course there are several other applications of WSNs in this field, but we give these applications as examples only.

1.7.3. Sensor Deployment

An important consideration when attempting to implement WSNs for use in any of the above applications is sensor deployment. In WSNs, a deployment can be deterministic, random, or a combination. The nature of the application and the requested performance are the main controllers of the shape of the deployment. For example, in a traffic light sequencing application, the deployment usually is deterministic as only a small set of sensors is needed to accomplish the task. Another reason for this determinism is that the scale is very small, i.e. limited to the intersection. On the contrary, a large scale deployment intended for use in accomplishing a surveillance task on a city scale would almost always be random. When the size of the WSN becomes very large, deterministic deployment becomes nearly impossible. Also, wireless sensor network design expects the

deployment to be random as the sensors may be deployed in some areas where human access would be dangerous, and also, because unattended operation is implied.

Several works have been proposed, as will be seen in Chapter 3, which suggest different deployment schemes some of which are deterministic and other non-deterministic. The design of an efficient deployment protocol requires considering several parameters such as the expected traffic pattern, geographical topology, application requirements, and system cost. We believe that such a design must be done in parallel with, and in close relation to, the routing protocol design. A good deployment should be assistive to the routing protocol, and a good routing protocol makes the highest benefit from the deployment and preserves it from loss due to node death as a result of fast energy depletion.

1.7.4. Routing in Wireless Sensor Networks

The main purpose of a WSN is to provide a specific service. In WSNs, all services occur by means of phenomenon sensing and data transfer. Therefore, it is necessary to assure the availability of an efficient routing protocol. In this field specifically, a huge amount of literature exists; each of the existing protocols tries to address a set of important parameters. For example, energy consumption, end-to-end delay, packet success rate, reliability, throughput, bandwidth, network lifetime, and service differentiation, are all very important parameters. Many classifications of routing protocols based on the network structure, protocol operation, quality of service, and data centricity have been proposed. Routing protocols are mainly concerned with routing the data packets from sources to the destinations while saving the limited node resources and maximizing the network's lifetime. Again, we believe that as separation between the

designs of deployment algorithms and routing protocols will lead to suboptimal performance. Therefore, we emphasize the importance of having a concurrent design for routing and deployment, and consider many routing related factors in the design of our deployment algorithm.

1.8. Issues and Challenges in Transportation Network Security

Although the necessary technology needed for ITS and transportation network security applications is available and even new technologies are being developed, there are still some problems that hinder or slow the implementation of these technologies. ITS America (2002) determined these challenges to be: 1) the trade off between security and privacy, as the integration of information systems usually tends to compromise privacy, 2) the negative effect on mobility, efficiency, and cost that may result from the implementation of these technologies if careful analysis, planning, and execution were not given the appropriate attention, 3) the desire for even better technologies and software tools to esnsure the prevention and detection of threats, 4) increase in technical complexity that is associated with the integration of these technologies with the already complex transportation system, and security issues that emanate from increasing the system size, 5) the resulting need to have additional layers of security to address the security of the surveillance technology itself, 6) funding, needed for developing, implementing, and maintaining these technologies, 7) institutional coordination challenges, and finally, 8) the missing community involvement through problem reporting and clarity of security needs.

One of the challenges that we find closely related to ability of implementing our proposed WSN deployment algorithm in reality is the privacy challenge. Although WSNs

were designed initially to have large numbers of sensors that are spread over a large geographical area to achieve some certain task, and that is what our algorithm does, there is a concern that given that large number of sensors, and the proposed deployment, it would be necessary to distribute these sensors in urban areas where there are thousands of personal properties. Community involvement by allowing the implementation of these devices within or near there properties raises a question about the feasibility of the whole process. Therefore, as indicated by ITS America, it is necessary to address these problems wisely to be able to make use of technologies that is expected to increase public safety.

CHAPTER II

PROBLEM STATEMENT

2.1. Modeling and Simulation Framework

This chapter describes the general modeling framework that unifies the problems under consideration and discusses each sub-problem in details. We start with the discussion of the framework and then go through each sub-problem individually. This will give a big picture of the targeted output and put the discussion of the problems in the appropriate contexts.

It was pointed out in the introduction that this work is concerned mainly with two parts; attacker-defender interaction and sensor deployment. These two elements are part

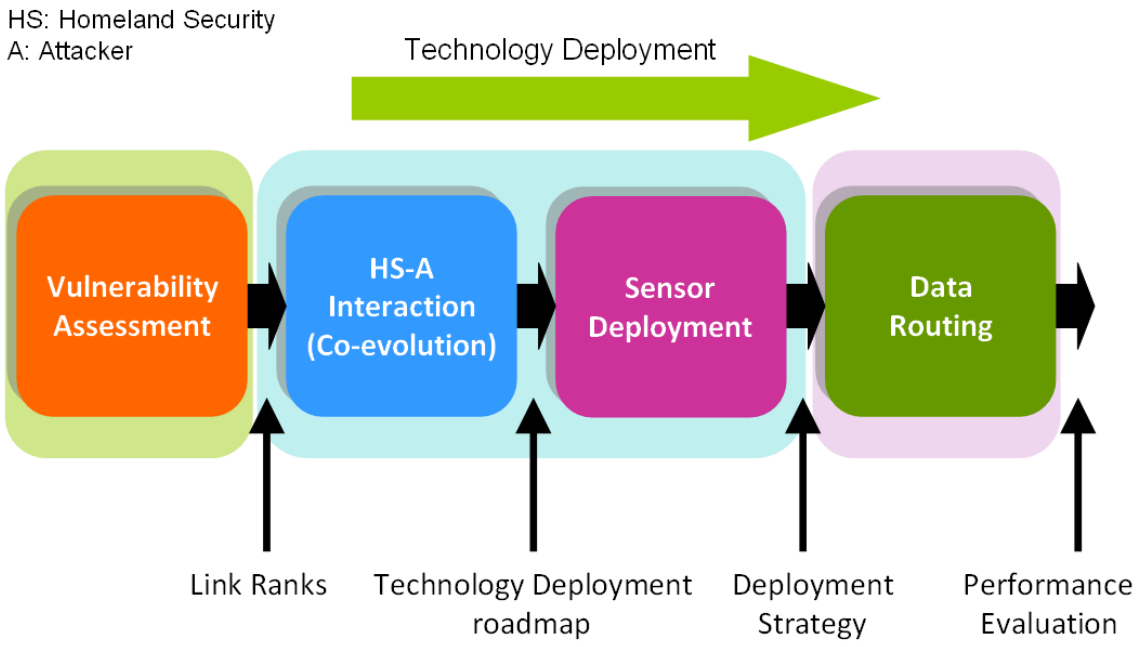


Figure3: Different elements of the used modeling and simulation framework

of a general framework that aims to assist an HS agency in making correct and informed decisions, in addition to assisting it in deploying a suitable technology for protection purposes. The framework consists of four elements as shown in Fig. 3. The two remaining elements are a vulnerability assessment model and a routing protocol.

The vulnerability assessment model (Wang et al. 2011) was proposed and implemented in Java previously by the research team. Although the details of that model are not within the scope of this work, we give a short description of its purpose and how it works. The objective of the model is clear from its name; it assesses the vulnerability of the considered system. The model takes, as input, a transportation network (road network) in the form of a text file containing trip origins and destinations, link free flow travel times, link capacities, origin-destination demand, and other variables. It uses Game Theory, where there are two players involved: an attacker (tester) and a defender (router), to assess the network's vulnerability in terms of the failure probability of each link. This is done through a game where the router tries to route traffic through the network and the tester tries to disable links. The routing probability for each link and the failure probabilities are calculated in every iteration of the game. User equilibrium is also considered in the traffic assignment model used. A convergence criterion is used to force convergence. At the end there is a set of link failure probabilities which can be easily mapped to link ranks. The output of this model is an input to the attacker-defender interaction model as shown in Fig. 3. Next, we formulate the problems for the two middle elements of the framework, and finally, end with the routing element.

2.2. Discussion of the First Sub-problem: Attacker-Defender Interaction

Referring to the middle section of Fig 3, it can be seen that after getting the desired information about the ranks of different roads in the considered network, the technology deployment planning process starts ahead. At this stage, it is necessary to know the available budget, the range of available and suitable technologies to select from, the time frame for deployment, and the suitable strategy for that deployment. The budget section of this process is again not within the scope of this work. In addition, we select WSNs as the protection technology to be used supported by the reasons discussed in the introduction. The remaining two sections are the deployment time frame and the strategy, which we will elaborate on.

Deployment timeframe is a very important element in the protection process. If the budget is constrained and an unwise deployment timeline is followed, this may lead to the drain of available resources quickly and lead to catastrophic consequences. Therefore, the objective of the first element, attacker-defender interaction model, is to study the interaction between the two entities over a timeframe in order to conclude a suitable deployment time plan.

Another issue addressed by the model is the estimation of the overall system vulnerability and the risk imposed on the system. Recall that the first element of the framework is the vulnerability assessment model. However, that element produces the link failure probabilities as the measure of vulnerability. On the other hand, the attacker-defender model produces the probability of attack success as a result of the interaction of all system elements. Therefore, the second model can be considered as the general

version that uses the individual link vulnerabilities to assess the overall vulnerability of the system.

Given the above discussion, the statement of the first considered problem can be formulated as follows:

Statement of Problem 1: "Given a set of transportation network links along with their respective ranks, which are based on the attack/failure probabilities of each link, produce a technology deployment roadmap indicating the timeframe and extension of deploying the technology and the increments (amount) of the deployment at each step. This roadmap should be based on the interaction between the HS agency and the attacker, considering the perceptions of both sides and the real world factors that affect that interaction. Also, provide an estimation of the actual probability of a successful attack on the network and the probabilities from the perspectives of each participating entity"

The solution of the above stated problem is accomplished by dividing it into three parts as will be seen in Chapter 4, and then integrating these parts in one big model. The modeling approach used is system dynamics, which was discussed earlier in the introduction.

2.3. Discussion of the Second Sub-problem: Sensor Deployment in Transportation Networks

The next step after preparing a deployment roadmap is to find a suitable deployment strategy. The purpose of this step is to find a deployment that guarantees the best performance of the system, facilitates the retrieval of the necessary information in a

timely manner, and extends the network's lifetime. As noted previously, this step is very closely related to the routing protocol step. A clear separation between the two can not be made without degrading the performance of the applied technology. Therefore, a consideration of each of these two elements while designing the other is unavoidable.

Two deployment algorithms were proposed in this work. Although the main target of such deployments is to properly protect the network, the difference arises from the capabilities of each deployment approach. One of the two approaches addresses routing-protocol related issues such as the high traffic near the base stations and the 'hole' problem only, while the other takes into consideration link ranks as well. The problem statement is provided below. It should be noted that the first proposed approach does not satisfy all the requirements, while the second does. This one of the reasons the second approach was proposed.

Statement of Problem 2: "Given a shapefile (GIS data file) containing network links and a set of respective link ranks, propose a deployment for the BSs, CCs, and the sensors that: 1) minimizes the effect of high near-base-station traffic on the depletion of node energy and the formation of holes around the BSs, 2) extends the network lifetime, 3) makes the data flow smoother throughout the network, 4) takes link ranks into consideration (highly ranked links will have higher data traffic around them; therefore, they should be given a special consideration similar to how the BSs and CCs are treated). The deployment must cover all network links. The nodes need to be deployed randomly and node densities at different locations in the network are to be output"

At the end of this stage, a ready to implement deployment should be available. However, a good design for the deployment strategy is not sufficient. The performance of

the resulting WSN is governed by the deployment as well as the routing protocol that will be used. A deficiency in one of them will hurt the overall performance. Hence, great attention should also be paid to the design of the routing protocol.

2.4. The Data Routing Problem

Referring back to Fig. 3, the forth and last element of the proposed framework is 'data routing'. The design of an efficient data routing protocol for the considered WSN is of immense importance. Even in the presence of a good deployment, if the routing protocol is badly designed, the influence on the performance is very severe. This may be in the form of long delivery delays, short network lifetime, low reliability, etc.

WSN routing protocols differ in design from one application to another. Application requirements such as node/sink mobility, high data rates, longer awake times, etc. pose restrictions on the design of a specific routing protocol. The common goals of all protocols are to provide high reliability, energy savings i.e. longer network lifetimes, low end-to-end delay, high delivery ratio, and other service and resource requirements, with differing levels.

Although the road was paved in this work to build an efficient routing protocol through the design of a routing-aware deployment, the design and performance testing of the protocol are considered as future work. Some work has been done in the design process, but it is not complete yet. Here, we present a statement for the routing problem to give the reader an idea about the features of the targeted protocol.

"Given the routing-aware deployment presented in this work, design a routing protocol that best utilizes the near-(BS, CC, and high-rank links) sensor redundancy to

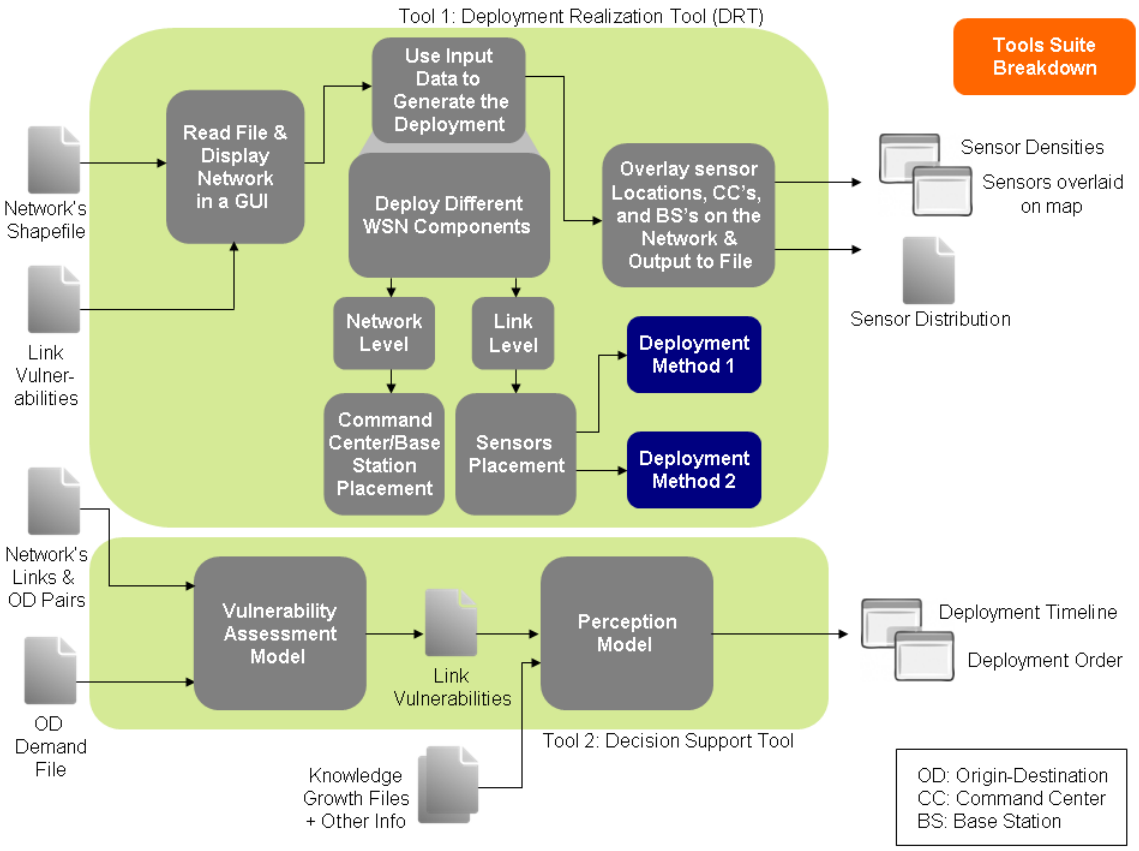


Figure 4: Tools suite

achieve: 1) minimum end-to-end delay, 2) longer network lifetime, 3) smoother data traffic flow through the network, and 4) 'hole'-introduction avoidance. The protocol must also have low control packet overhead and simple computational complexity"

2.5. Built Tools and their Integration

The development of the different elements of the proposed framework is accompanied by building a suite of homeland security assistive tools for decision support, sensor deployment, and data routing performance evaluation. Two of these tools, namely: the decision support and the sensor deployment tools, have already been implemented (only the sensor deployment tool has a GUI currently). The decision support tool consists of the

first two elements of the framework, i.e. the vulnerability assessment model and the attacker-defender interaction model (also called perception model). The third tool; routing protocol performance evaluation tool, is still under development. The usage flow, inputs, and outputs of these tools are shown in Fig. 4.

CHAPTER III

PREVIOUS WORK

3.1. Attacker-Defender Interaction

The first part of this work is concerned with analyzing the risk of having a successful transportation network attack, and modeling the behavior of an attacker in response to the efforts of intelligence agencies to counter his trials to attack the system by means of technology deployment. Risk analysis is a wide field that is being used in different disciplines such as management science, health care, and security. Several quantitative and qualitative models to analyze and assess risk have been made. These models have been suggested in different contexts. The common target of all them is to analyze risk. We limit our attention to terrorism risk analysis in this work.

Qualitative risk analysis has been in use for a long time. Researchers and risk analyzers prefer using this type of analysis because it does not require knowledge of the probabilities of different events and is easier to perform. It depends mainly on the rating of different parameters and on expert opinions. There are several works that use this approach, which is really helpful in many contexts. For example, Hudson (1999), Crenshaw (2000), Borum (2004), Victoroff (2005), and Ackerman et al. (2007), all focused on surveying existing psychological and sociological theories, classifications of terrorist mind-sets, behaviors, motivations, and on analyzing available data about these terrorists. In a similar context, Jackson et al. (2007) studied the interaction between defensive technologies and counter measures and their co-evolution on four terrorist

groups. They, then, drew a group of lessons for technology planning and the design of defensive technologies.

The other approach, quantitative risk analysis, was followed by many other researchers. Although it has some difficulties associated with it, like acquiring the probabilities of different attacks, it is sometimes worth doing since it provides us with quantitative measures that can be used to take more accurate decisions. Howard (2005), for example, proposed a quantitative measure to assess the usefulness of the used security measures in deterrence methodologies for the commercial ferry attack scenario. The measure represented a "cognitive state in the mind of the terrorist" which accounted for the perceived observable quality of the protection measure, its perceived accuracy, and its perceived reliability. Jenelius et al. (2010) used the Game Theory approach to find the optimal resource allocation based on system element ranks. They formulated the attacker and defender problems as optimization problems, and considered many factors such as the observation errors on the attacker side, uncertainties on the HS side, many system elements, different attacker types, and the non-attack alternative. A system for decision support, threat response planning, and risk assessment was proposed by Tsang et al. (2010). They used Agent-based simulation, swarm intelligence, social science findings, and real humans to conduct the experiments in a virtual environment. They also considered crowd behaviors, terrorist attacks, and rescue missions. As a trial to evaluate defender's investments, Parnell et al. (2009) used the defend-attack-defend decision analysis model. They also provided a comparison between uncertain hazard risk analyses with intelligent adversary risk analysis. In a slightly different context; computer system security, Almasizadeh and Azgomi (2008) used Markov Chains (MC) to model the attack

process on a computer system. Intrusion process was considered as atomic sequential steps, and the system tried to transfer its state to a secure state after each atomic step. We believe this technique can be generalized to other attack scenarios. Paté-Cornell and Guikema (2002) presented a probabilistic model that addresses the objectives of the U.S. and the attacker, the different possible scenarios, and the dynamic interaction between the two opponents. They used risk and decision analyses as well as some game theoretic elements to build their model. Their target was to provide a reasoning approach to help prioritizing the threats and the selection of appropriate countermeasures. The study of benefits of the countermeasures in the reduction of threats was also a main goal. Another piece of work, by Bell et al. (2008), used the game theoretic approach to posit attack or failure scenarios on the road network, and to minimize the maximum expected loss if the event occurs. Insua et al. (2009) and Rios (2010) suggested a framework for Adversarial Risk Analysis (ARA) that extends traditional risk analysis techniques and included statistical risk analysis, Game theory, and Bayesian analysis. They, in addition, surveyed counter-terrorism models such as: the sequential defend-attack model, simultaneous defend-attack model, defend-attack-defend model, and sequential defend-attack model with defender's private info.

Finally, we conclude with a classification of risk assessment techniques as proposed by Cummings et al. (2006). They classified the existing risk analysis techniques into standard and emerging ones. Some of these techniques are shown in Fig. 5. Furthermore, we show the components of Insua et al.'s proposed framework on the figure. As we use system dynamics for risk analysis in this work, it is necessary to show where it belongs in

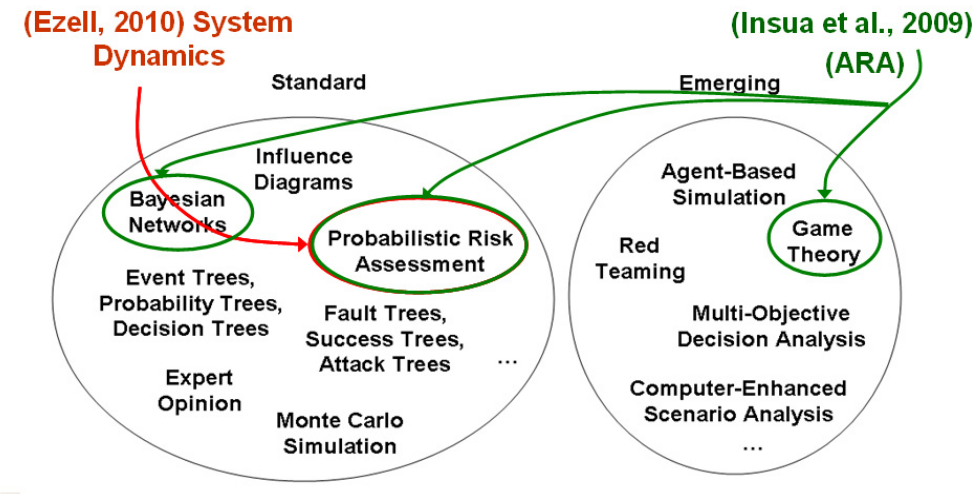


Figure 5: Risk Assessment Techniques

that classification. Ezell et al. classified system dynamics as belonging to probabilistic risk assessment.

3.2. Wireless Sensor Network Deployment in Transportation Networks

Transportation network security has gained special attention in the recent years due the repeated and, in several cases, successful attack trials. Several techniques are used to achieve this security, some of them are technology-based and others are strategy, planning, and design based. Dornan and Maier (2005) listed four categories for security incident countermeasures in surface transportation; prevention, protection, redundancy, and recovery. As stated in their report, prevention includes the means used to disallow access to some sensitive assets of the transportation system by the use of either technology, such as intrusion detection systems, CCTV's, and access control systems, or by non-technology based methods like fences, doors, locks etc. Protection is used when prevention is insufficient and includes target hardening by suggesting new designs or

retrofitting old ones. The third category, redundancy, is meant to be on the modes' level, by having redundant transportation modes and redundancy in each mode, and on the technology level used to improve the security, such as redundant data exchange methods, to avoid failure in the case of an incident. Recovery is concerned with disaster response systems such as emergency notification systems and first response as well as rerouting of services and reconstruction. Wireless sensor networks can be used in any of these categories. Applications of WSNs include intrusion detection (prevention), structure condition monitoring (protection and target hardening), monitoring of large scale areas for different events (technological redundancy), and incident location monitoring and assessment of the incident's severity (recovery).

Y. Chen et al. (2009) proposed a transportation network specific WSN deployment and an associated routing protocol. The primary focus was on the routing protocol development. For the deployment part, they used a system architecture consisting of three elements; a Mobile Sink (MS), Sensor Nodes (SN), and Vice Sinks (VS). The mobile sink is carried by a vehicle moving along the road. Sensors nodes were assumed to be densely deployed in the sensing field and are responsible for routing the sensed data from one VS to the other. VSs were assumed to be fixed on light posts, have unconstrained energy, and are the only nodes capable of communicating the data with the MS. The architecture is shown in Fig. 6. Although the authors addressed energy efficiency in the design of the routing protocol, we believe that the architecture used does not address the fast death issue in the nodes lying in the vicinity of the road from a deployment perspective. These nodes will be more susceptible to energy depletion due to the higher packet traffic near the road; as any traffic moving towards the sensor field or leaving it

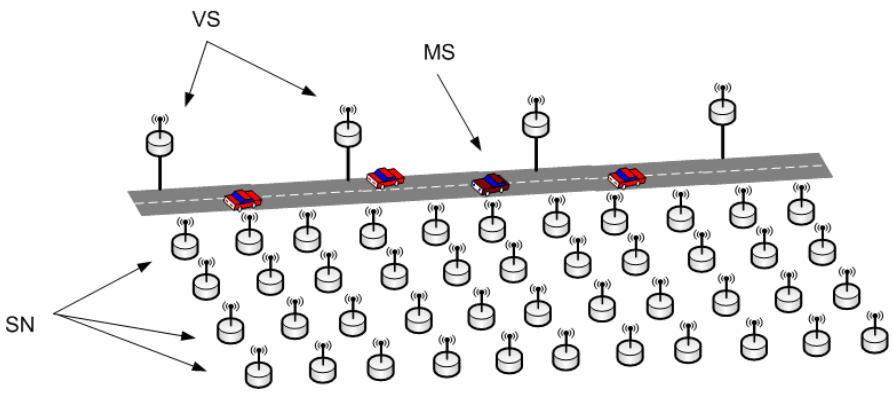


Figure 6: The deployment used by Y. Chen et al. (SN: sensor node, VS: vice sink, MS: mobile sink)

towards the VSs will be through them. Also, the installation of fixed nodes along the roads is not scalable for a large number of roads with many of them having a high priority. Finally, this architecture relies on the MS and is not suitable for the surveillance scenario that we consider; where continuous monitoring and timely detection is required.

Yuan and Zhu (2006) proposed a wireless sensor transportation monitoring network (WSTMN) that consists of a mixture of both wired and wireless network components. The wireless part was a sensor network that had five types of nodes: monitoring center node, gateway nodes (BSs), sink nodes, sensor nodes and target nodes, according to their naming criteria. In their suggested scheme, sensor nodes were assumed to be placed in the road surface. They also found the minimum number of sensors needed to provide the monitoring capabilities while minimizing the system cost. Fig. 7 shows an illustration of their proposed system. Again, for a large scale deployment that covers the road network of a whole city, it is very difficult to have such manual and deterministic deployment. Also, placing such sensors in the road surface would be very expensive on the large scale.

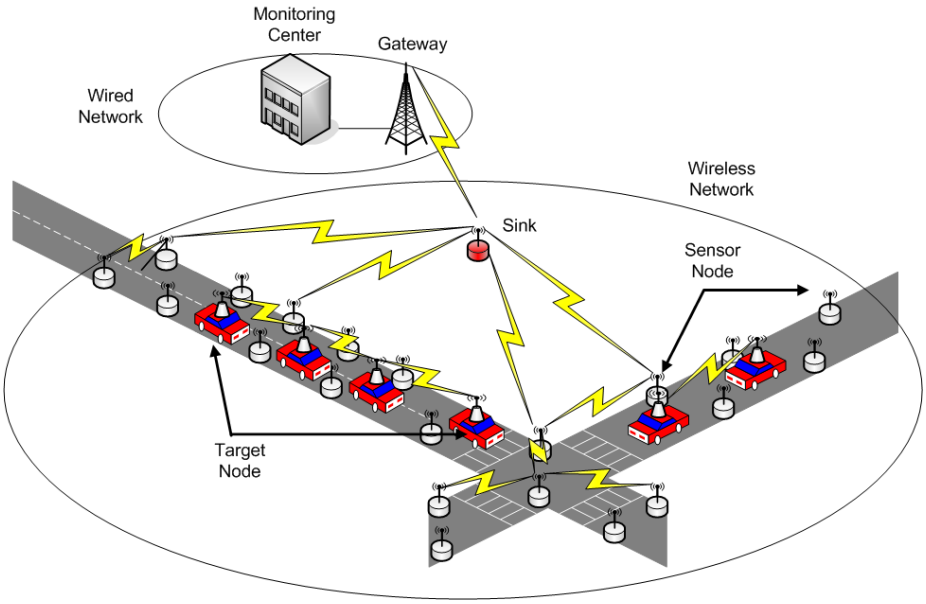


Figure 7: The deployment proposed by Yuan and Zhu

An interesting and important research done by Cheung and Varaiya (2007) as part of the California PATH program studied and experimented the use of WSNs in transportation network surveillance. They conducted several experiments and analyses in the context of vehicle detection, classification, and reidentification. However, all these experiments were conducted using a very small number of sensors and the deployment was always deterministic. It is worth to mention, however, that they pointed out the benefits of having a large scale deployment of WSN due to its capabilities and configuration flexibility. Also, the possibility of integrating WSNs in multi-function wireless ITS systems, that not only do surveillance but also can sense road conditions and assist vehicle-infrastructure integration (VII) techniques, was pointed out.

Wenjie et al. (2005) proposed an architecture based on WSNs for guidance and control in ITSs. The objective of their work was to minimize the average travel time of the vehicles in the network. They considered a traffic system consisting of three main

components; intersections, roads, and vehicles. The system had three types of WSN nodes corresponding to the above elements, i.e. an intersection unit, roadside unit and vehicle unit. Roadside and intersection units were assumed to be fixed on light posts, while vehicle units are carried by each vehicle. Again, the deployment was deterministic. Although this kind of deployment may serve the objective if their work, we do not believe it is suitable for the attack detection and prevention context for the reasons mentioned previously. Fig. 8 shows their suggested architecture.

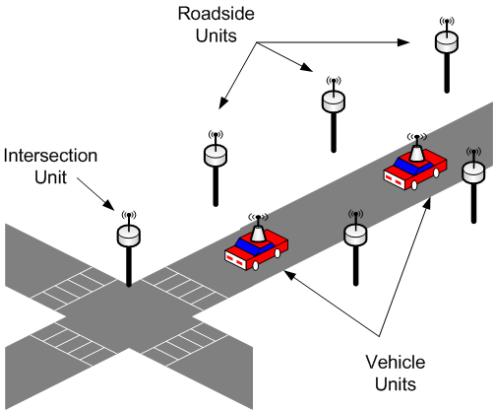


Figure 8: The network architecture proposed by Wenjie et al.

Mirko et al. (2009) introduced a traffic monitoring system based on WSNs. The architecture of the system included three basic elements. The first is a Road Side Unit (RSU) that is responsible for retrieving data from a Gateway Node (GN), which is the second element, and performing advanced data processing. The connection between the GN and RSU is wired, and the RSU can be connected to more than one GN. The third element is a set of n Sensor Nodes (SNs) that collect traffic and road condition data and forward it to the GN. The proposed deployment is linear along the roadside with regular spacing between sensor nodes. Even the RSU, as the name indicates, is on the side of the

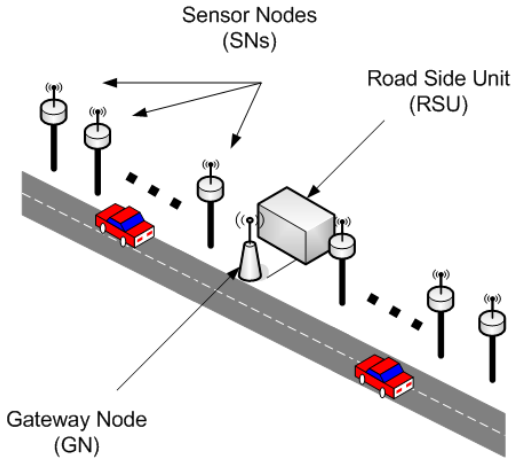
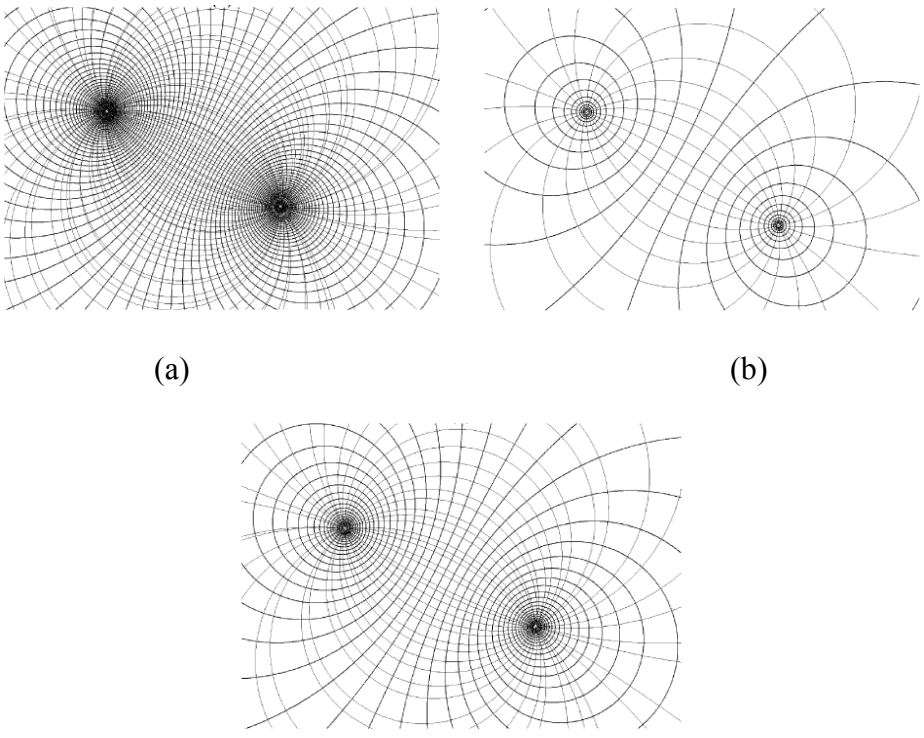


Figure 9: The network architecture proposed by Mirko et al.

road. The system is intended for collecting traffic data that can be used for generating safety warnings. Fig. 9 shows the architecture of the proposed system.

We conclude with a very interesting piece of work by Toumpis and Tassiulas (2006). They proposed an approach for optimal deployment of large WSNs. The approach depended mainly on the big similarity between the features of WSNs and electrostatic fields. The sources were represented by positive charges while the sinks were represented as negative charges. Instead of dealing with the deployment problem on the microscopic scale - i.e. trying to find locations of individual sensors - they followed a macroscopic approach by considering deployment densities at different locations. An assumption that the network considered for deployment is highly dense, and that the sources and the sinks of information are spatially distributed were used in their proposal. After approximating the sources and the sinks by an information density function, they formulated the deployment problem as a constrained optimization problem. The solution of that problem produced the optimal spatial density of sensor nodes, their total number, and the induced



(a) (b)

(c)

Figure 10: Electric field lines and constant potential loci created by two opposite charges. (a) q = 1 Cb. (b) q = 2Cb. (c) q = 4Cb. (Courtesy of Toumpis and Tassiulas, 2006)

traffic. Eventually, the nodes (according to the found number) are placed with a node density function as close as possible to the optimal density function. In one of two cases discussed, where a bandwidth-limited capacity achieving physical layer was used, the final step of placing sensors can be done by placing a sensor at the intersection of each electric field line with the constant potential loci. An example of these field lines, constant potential loci, and their intersections that are used for sensor placement is shown in Fig. 10. The above work is very closely related to the deployment algorithms proposed here. Our work follows a very similar approach in terms of the use of electrostatics and the sensor density based deployment.

CHAPTER IV

METHODOLOGY

4.1. Introduction to the Solution Approach

The first part of the work uses a mixture of prescriptive and descriptive models to tackle the problem of the evolving interaction between HS agencies and attackers. We believe this should always be the case with such models, as their objective is to help HS agencies make informed decisions. Therefore, it is necessary to account for the behaviors and actions of the attacker and then be able to draw conclusions based on these behaviors and actions to use them in making correct decisions. Accounting for attacker characteristics/actions requires a descriptive model that maps them to an accurate computer model that correctly simulates them. On the other hand, drawing conclusions and getting guidance about which decisions are appropriate for a specific situation requires the use of a prescriptive model. Hence, we add a prescriptive component to the model represented by the variables aimed at guiding the HS personnel to make decisions. In addition to that, another descriptive sub-model is responsible for describing real-world-related factors and events that affect the interaction. The overall objective of the model, with its prescriptive and descriptive components, is to assist HS agencies in planning for the deployment of a suitable protective technology. This is obtained based on the knowledge of the probable attacks that could happen in some time frame. Another goal is to give them an idea about the vulnerability of the transportation network. System dynamics is utilized to build the model as it enables the inclusion of numerous model variables and the specification of their interrelations in a straightforward manner. This

makes it easier for the modeler to focus on the proper design of the model. Another advantage is that system dynamics facilitates the conduction of sensitivity analyses to study the influence of different factors affecting the interaction.

In the second part, the approach is mainly based on Geographic Information Systems (GIS) data and computational geometry. The problem of sensor deployment is application specific. It is nearly impossible to find a single deployment strategy that fits all applications. As we selected transportation network links as our application context, it was necessary to find a realistic approach for deploying the sensors. It is not possible to have a network-wide deployment for the purpose of testing; this would be very expensive and impractical. Therefore, it was concluded that a modeling and simulation approach would be the best choice, especially for city scale we consider. The high level summery of the approach is as follows: First, a shape file (which is one form of representing GIS data such as city borders, road network links, buildings, etc.) is used to read the road network links; Second: some computational geometry operations take place on the read data; Finally, two suggested sensor deployments are output in the form of shape files containing the distributions of sensors and the locations of the Base Stations (BSs) and Command Centers (CCs) based on a specific coordinate system. Outputs are also provided in the form of deployment radii, grid cell center locations, and sensor densities in a text format.

4.2. Detailed Discussion of Sub-problem Solution Approaches

In this section, we discuss in details the selected solution approaches for the two main parts of the work. We start with the attacker-defender interaction model, where we present the model specification, then the model is formulated and the underlying

equations are provided and discussed. Followed by that is the discussion of the sensor deployment approach. In that subsection, we first explain the main idea behind the selection of the proposed approach, and then provide the details of the deployment and all constituent algorithms.

4.2.1. Attacker-Defender Interaction

This subsection presents the model specification, the underlying equations of each sub-model, and the detailed explanation of the rationale for and the meaning of each variable.

4.2.1.1. Model Specification:

The attacker-defender interaction model, also called the Perception squared Interaction cubed P^2I^3 model (Tolba et al., 2011) or simply the perception model, utilizes system dynamics as the underlying modeling approach. This enables the modeling of the complex interactions between the homeland security (HS) agency and the attacker. This approach depends mainly on differential equations where rate variables control the levels of the dependent variables. VenSim™[1] was used as the platform of the simulation. In the proposed model it was assumed that the attacker targets transportation network links only. This will be extended to other network elements in future work. A rational attacker was also assumed. As the model is intended to provide the infrastructure protection (HS) agency with a time plan for technology deployment, the considered time period was set to 10 years as a reasonable period for a medium- to large-sized networks (hundreds to few thousands of links). Any other length for that period would have worked equally well.

[1] http://www.vensim.com/

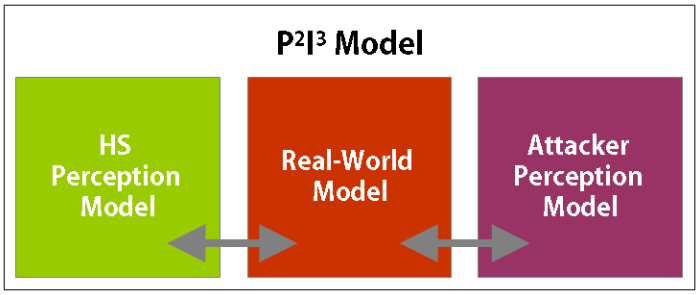


Figure 11: Attacker-defender interaction model's block diagram

The model consists of three sub-models: the HS-perception model, the attacker-perception model, and the real-world model. These three models are interconnected and unite to form the attacker-defender model. Fig. 11 shows the model's block diagram. The model defines three different probabilities of attack success. Two of them represent the probabilities based on the perceptions of the two opponents. The third is assumed to be the actual probability based on their actions and on other uncontrolled factors. Finally, none of the two participants is assumed to have complete information about the accomplishments of the other. Table 1 lists the abbreviations used from here on for the frequently repeated terms of the model.

Table 1: Model Abbreviations

Abbreviation	Meaning
HS-S	Homeland Security Seen
HS-P	Homeland Security Perceived
HS-A	Homeland Security Assumed
HS-T	Homeland Security Tolerable
A-S	Attacker Seen
A-P	Attacker Perceived
ACT	Actual

4.2.1.2. Model Operation, Variables, and Equations:

Model operation, variables, and equations are explained in details in this subsection. This is first done for the HS perception model, then for the attacker perception model, and finally, for the real-world model. In order to make the discussion clear and useful, we classify the variables used in the attacker-defender model into three categories: mixed variables, pure sub-model variables, and interaction (simulation) variables as shown in Fig. 12. Mixed variables are shared among two or more sub-models. Pure model variables only belong to one of the sub-models. The variables in the third category belong to the interaction itself, such as its time period. In addition, in the following discussion, we point out each model's indicator, input, and output variables.

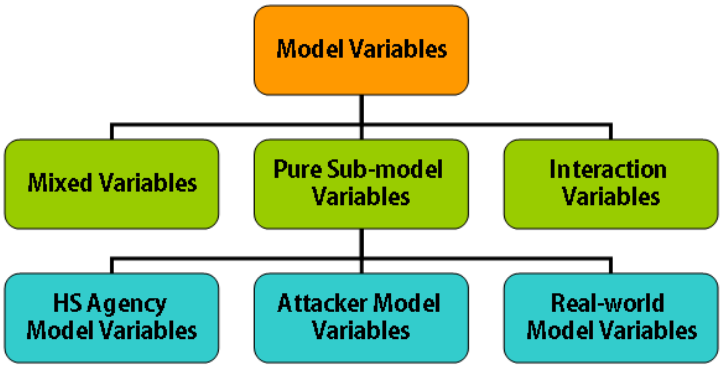


Figure 12: Categories of the variables used in the HS-Attacker interaction model

a. Homeland-Security-Perception Model:

The first sub-model addresses the perception of the HS personnel, the resulting decisions, and the factors affecting these decisions. The flow chart of this model is shown in Fig. 13. The operation of the model will be discussed through its variables with the assistance of the flow chart.

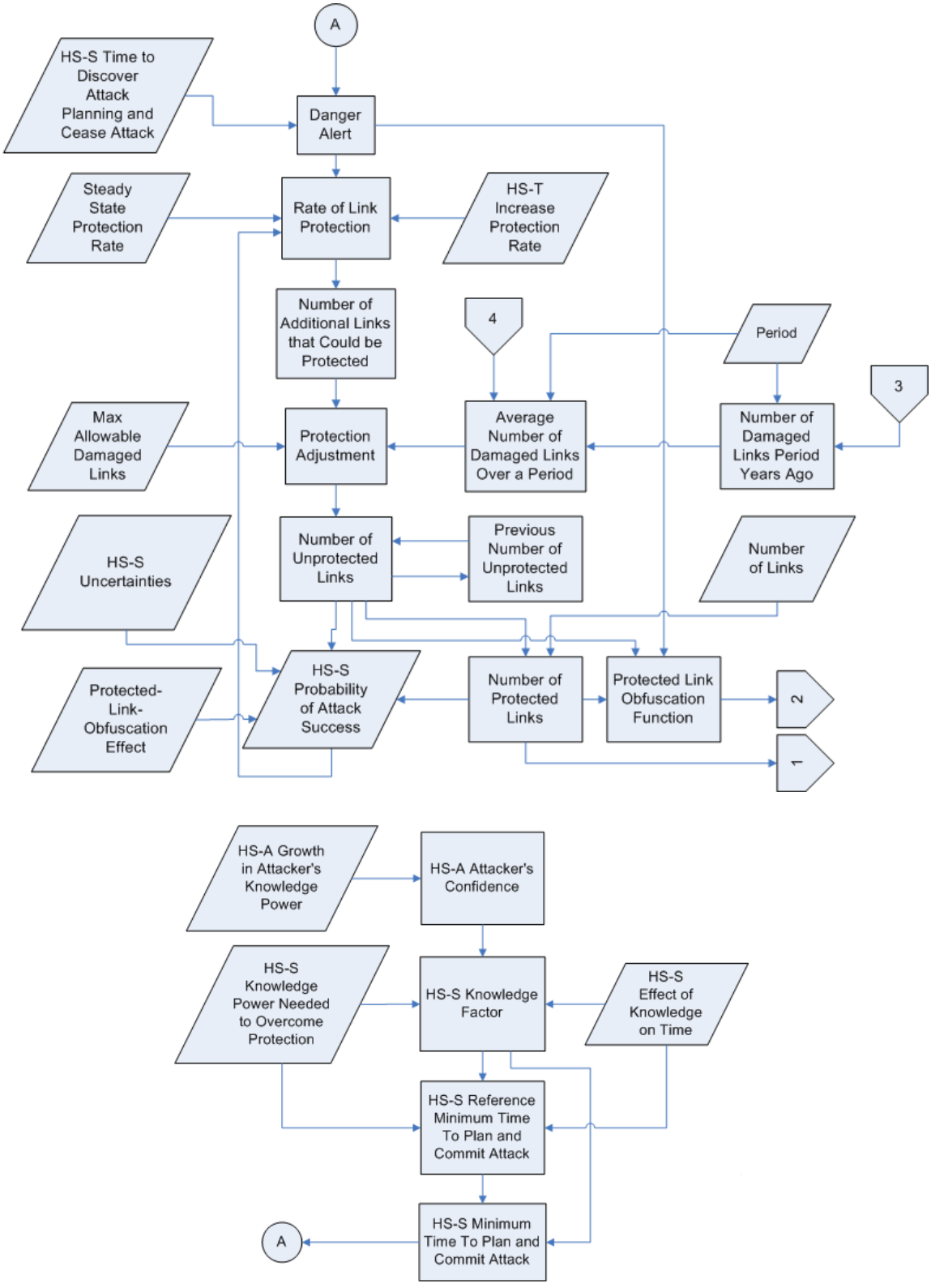


Figure 13: HS-Perception Model flow chart

HS has its own view of the probability of attack success. This may be significantly different from the A-S probability and/or the actual probability of attack success. In this sub-model, the HS Assumed (HS-A) growth rate in attacker's knowledge affects the HS Perceived (HS-P) amount of information that the attacker gains over time. We are assuming that knowledge always increases as this is the natural case except in some rare situations. As the amount of information the attacker has gained exceeds an HS-specified amount, and based on the HS-A effect of gaining this information on the time needed to plan and commit an attack, a danger alert can be fired. If this happens, a modification is made to the steady-state link protection rate. Then, based on the maximum tolerable number of damaged links in the period of study, a protection adjustment is made. This in turn, changes the number of links that are protected, which affects the probability of attack success from the HS agency's point of view. This loop keeps updating itself over the period of study until all the links in the network become protected. Although the model runs until all links are protected, it provides the HS agency with graphs of the probability of attack success and number of protected links versus time, which can be used to stop at an acceptable probability of attack success instead of protecting all links. Budget and technology deployment feasibility govern this decision. This model gives a planning road-map that tells the HS agency when to change the protection deployment rate and when to keep it constant over some period of time. This will achieve the highest benefit when the HS-A (based on intelligence procedures and on experience) rate of growth in attacker's knowledge is very close to the actual one. It also provides the probability of attack success as could be seen by the HS agency as a function of all the data collected by that agency and the preparation done up to the time of conducting the

simulation. The main input to this model is the "HS-A Growth in Attacker's Knowledge Power". This differs from the variable "Growth in Attacker's Knowledge Power" that appears as a joint variable in the attacker-perception model and the real-world model in how HS acquires it. Although both rates are assumed, there is an inherent difference between them. The "HS-A Growth in Attacker's Knowledge Power" is obtained based on the intelligence information the HS agency has collected about the attacker and the environment by the time the simulation is conducted. We call it assumed because the information collected by the HS agency is never complete. So, there is an assumption component in this rate. The other rate can be considered as a reference or as the rate HS believes is the actual rate of growth in attacker's information both in the attacker's perception and in reality (it is also considered the actual rate for simplicity, although it can be split into two). So, we are trying to answer the question:

"If there is a rate that the attacker thinks is the rate he can gain information at in some period of time, and HS has its perceived rate of growth of this information based on collected data, what is the chance that the attacker will succeed, from HS's perspective, from his perspective, and in reality if these assumptions are true?"

The unit "BitsOfInfo/Year" is used to express this growth rate. For simplicity, we expressed the confidence of an attacker in terms of the "BitsOfInfo" he has. Therefore, the variable HS-A Attacker's Confidence is no more than the integral of this rate over the study period. The HS-S Knowledge Factor represents the effect of the difference in knowledge level between that needed to overcome protection from HS's point of view and that of the attacker on the time of planning and committing an attack. This is based on an assumed fixed amount of reduction in the needed time per a bit of information

increase as expressed in the variable "HS-S Effect of Knowledge on Time". The HS-S Knowledge Factor changes the reference amount of time required to plan and commit an attack (which is the initial value for that variable based on intelligence information). If this time becomes less than the time the HS agency knows is required to discover and cease attack, the danger alert is fired. After the danger alert is fired, based on whether there is a steady state rate that the HS agency uses for deploying technology, and based on the HS Tolerable (HS-T) increase in this rate, the number of additional links that can be protected is decided. Of course, the tolerable increase in the protection rate is governed by budget in real life. The maximum allowable number of damaged links over the period of study determines the fraction of the protection adjustment calculated that will be actually used. The used fraction specifies how many additional links will actually be protected. The most important variable in this sub-model is the HS-S Probability of Attack Success. It depends on four variables: the HS-S Uncertainties; which represents the effect of incomplete information on the probability of attack success, the numbers of protected and unprotected links, and benefit of adding ambiguity to the deployed technology so that it becomes non obvious to the attacker whether a specific link is protected or not (this is represented by the variable O_e: Protected-Link-Obfuscation Effect). The following list summarizes the types, meanings, and the governing equations of each variable used in this sub-model.

- *HS-A Growth in Attacker's Knowledge Power*, $G_{RK}^{HS-A}(t)$: The main input in this sub-model. It should be based on intelligence information collected by the HS agency and is measured in "BitsOfInfo / Year". It is therefore expected to be input as a set of values representing that variable growth rate over the time period under consideration.

- *HS-A Attacker's Confidence*, C_A^{HS-A}: This is a level variable measured in "BitsOfInfo" and is intended to represent the HS agency's perception about the amount of information the attacker is going to gain over the technology deployment period. Eq. 4 expresses attacker's confidence as a function of the growth rate in his knowledge.

$$C_A^{HS-A} = \int_0^t G_{RK}^{HS-A}(t)dt \tag{4}$$

- *HS-S Knowledge Power Needed to Overcome Protection*, P_K^{HS-S}: A constant reflecting the amount of information the HS agency believes is sufficient to overcome the deployed protection. It is measured in "BitsOfInfo".
- *HS-S Effect of Knowledge on Time*, E_K^{HS-S}: A constant representing the reduction effect that gaining information (by the attacker) has on the time needed to plan and commit attack.
- HS-S Knowledge Factor, F_K^{HS-S}: A factor representing the additional time added (to the minimum time needed to plan and commit attack if the attacker has all the desired information to attack) if there is still some information that the attacker needs to acquire to be able to overcome the deployed technology. This factor is the difference between the desired information to overcome technology and the information the attacker has, all multiplied by the expected amount of reduction in desired time per bit of information. Eq. 5 shows this relation.

$$F_K^{HS-S} = (P_K^{HS-S} - C_A^{HS-A}) \times E_K^{HS-S} \tag{5}$$

- HS-S Reference Minimum Time To Plan and Commit Attack, $\hat{T}_{PCA}^{HS-S}$: The time needed by the attacker (in HS's perception) for preparing a plan and executing it if he has the desired information to overcome the deployed technology. It is measured in "Years"

and is computed based on the amount of information needed to overcome the protection and the reduction in needed time per bit of information as given in Eq. 6.

$$\hat{T}_{PCA}^{HS-S} = E_K^{HS-S} \times P_K^{HS-S} \tag{6}$$

- HS-S Minimum Time To Plan and Commit Attack, T_{PCA}^{HS-S}: This variable represents the minimum time needed by the attacker in HS agency's perception to plan for and commit an attack. If the attacker has the desired information to overcome the deployed technology (in HS's perception) or even more information then there must be a minimum time to create the plan and actually execute the attack. On the other hand if the attacker doesn't have the necessary information then this time gets longer. This variable is measured in "Years" and is expressed by Eq. 7.

$$T_{PCA}^{HS-S} = \begin{cases} \hat{T}_{PCA}^{HS-S} & , F_K^{HS-S} < 0 \\ \hat{T}_{PCA}^{HS-S} + F_K^{HS-S} & , otherwise \end{cases} \tag{7}$$

- HS-S Time to Discover Attack Planning and Cease Attack, T_{DPCA}^{HS-S}: A constant that represents the expected amount of time needed to discover planning for an attack and ceasing the attack. It is measured in "Years".

- Danger Alert, A_d: An dimensionless indicator variable that is intended to represent an alarm or a strategy change motivator for the HS agency. When the minimum time to plan and commit attack becomes less the time needed by the HS agency to discover attack planning and cease that attack, this alert is fired; i.e. it takes a value of 1, otherwise it is zero. It is shown in Eq. 8.

$$A_d = \begin{cases} 1 & , T_{PCA}^{HS-S} \le T_{DPCA}^{HS-S} \\ 0 & , otherwise \end{cases} \tag{8}$$

- HS-T Increase Protection Rate, R_{IP}^{HS-T}: A constant specifying the affordable increase in the steady state number of links that get protected per year without excess burden on the HS agency. The unit for this variable is "Links/Year".
- Steady State Protection Rate, R_{SSP}: A constant reflecting the regular rate at which the HS agency has been protecting links each year. It could be set to zero if no rate was used before.
- Rate of Link Protection, R_{LP}: The compound rate of link protection in "Links/Year". This rate is the steady state rate added to which the tolerable increase in the rate conditional on an alert and is controlled by the current value HS perceived probability of attack success. This variable serves as a first level control over the number of links that are actually protected. The second level is the maximum tolerable number of damaged links over a period of time if any. Eq. 9 illustrates this relation.

$$R_{LP} = P_{AS}^{HS-S} \times ((A_d \times R_{IP}^{HS-T}) + R_{SSP}) \tag{9}$$

- Number of Additional Links that Could be Protected, N_{AL}: This is a level variable controlled by the rate of link protection previously defined. It is an integration of that rate over the technology deployment time period. This associated unit is "Links". This is shown in Eq. 10.

$$N_{AL} = \int_0^t R_{LP}(t)dt \tag{10}$$

- Maximum Allowable Damaged Links, N_{DL}^{*}: A constant specifying the maximum number of links that the HS agency can accept to be damaged over the deployment period. Of course it could be set equal to zero if no tolerance is allowed. There is also an upper bound on this number that can be specified in the model.

- Protection Adjustment, A_P: The number of links that will actually be additionally protected. This is controlled by the ratio the average number of damaged links over a pre-specified previous period of time and the maximum tolerable number damaged links over the deployment period. Eq. 11 describes this relation.

$$A_P = \begin{cases} \lfloor N_{AL} \rfloor & \hat{N}_{DL} > N^*_{DL} \\ 0 & otherwise \end{cases} \tag{11}$$

- Average Number of Damaged Links Over a Period, $\hat{N}_{DL}$: The average number of links that are actually damaged over a pre-specified period p (measured in "Years") of time. This is given by Eq. 12 and 13 as the difference between the current and the p-years ago number of damaged links divided by that period.

$$\hat{N}_{DL} = \frac{(N_{DL} - N^P_{DL})}{p} \tag{12}$$

$$N^P_{DL} = N_{DL}(t - p) \tag{13}$$

- HS-S Uncertainties, U^{HS-S}: A dimensionless constant greater than zero and less than or equal to one that reflects the effect of incomplete information on HS's side on its perceived probability of attack success.

- Protected-Link-Obfuscation Effect, O_e: An input constant that represents the expected effect of the utilization of some protection technology obfuscation technique on the probability of attack success. It is a positive integer lying in a pre-specified interval of values associated such effects (taken to be form 1 to 10 in this model). The value of this constant should be based on the collected information about results of obfuscation efforts done by the HS agency on the behavior of the attacker.

- Number of Unprotected Links, N_{UP}: This variable measures the number of unprotected links in the network. It is either equal to the previous year's value if there is

no need to protect more links or its value is decreased if an adjustment is done to the protection (i.e. more links will get protected this year). This is of course conditional on the presence of unprotected links. Eq. 14, 15, and 16 show these relations.

$$N_{UP} = \begin{cases} \alpha & N_{UP}^{p} > 0 \\ 0 & otherwise \end{cases} \tag{14}$$

Where:

$$\alpha = \begin{cases} \beta & A_P > 0 \\ N_{UP}^{p} & otherwise \end{cases} \tag{15}$$

$$\beta = \begin{cases} N_{UP}^{p} - \lfloor A_P \rfloor & \lfloor A_P \rfloor \leq N_{UP}^{p} \\ 0 & otherwise \end{cases} \tag{16}$$

- Previous Number of Unprotected Links, N_{UP}^{p}: Specifies the previous year's number of unprotected links. Initially, it is specified by the input variable N_{UP}^{i}. The equation of this variable is given in 17.

$$N_{UP}^{p} = \begin{cases} N_{UP}^{i} & t = 0 \\ N_{UP}(t-1) & otherwise \end{cases} \tag{17}$$

- Number of Protected Links, N_P: Is given by the difference between the total number of links NL, which is an input, and the number of unprotected links as given by 18.

$$N_P = N_L - \lfloor N_{UP} \rfloor \tag{18}$$

- HS-S Probability of Attack Success, P_s^{HS-S}: An output variable representing the probability of attack success perceived by the HS agency based on the previous calculations in the above variables. It increases when the number of unprotected links is a large fraction of the total. When the protected link obfuscation effect is large, the

probability decreases. Finally, if the HS agency is highly uncertain about the information it has about the attacker, the probability become very high. This is expressed by Eq. 19.

$$P_s^{HS-S} = \frac{N_{UP} \times U^{HS-S}}{(N_P + N_{UP}) \times O_e} \tag{19}$$

- Protected Link Obfuscation Function, O_f: An input variable that models the effect of deploying some illusive devices, e.g. empty black boxes, to deceive that attacker and make him think that some links are protected while they are not. This variable is expressed as a function of (an operator on) the number of protected, unprotected links, and the danger alert. The danger alert is incorporated to account for the required increase in this obfuscation when the HS-S minimum time to plan and commit attack becomes less than the time needed to discover that attack. The form of this function should be based on the experiences of the HS agency and its experiences. It is one of the difficulties HS agency would face as it is required to find a proper formulation that reflects the actual effect of deception techniques used on the attacker. In the simulation, an arbitrary function was used. This unit used to measure this variable is "links" as the output of the function should represent the number of links the attacker will perceive as protected links. Eq. 20 shows this functional relation.

$$O_f = f(A_d, N_{UP}, N_P) \tag{20}$$

b. Attacker-Perception Model:

The attacker has his own view and factors that affect his decisions. A very similar set of variables to the ones used in the HS-perception model exist on the attacker's side, but this time they represent the attacker's point of view. They reflect the data that is collected

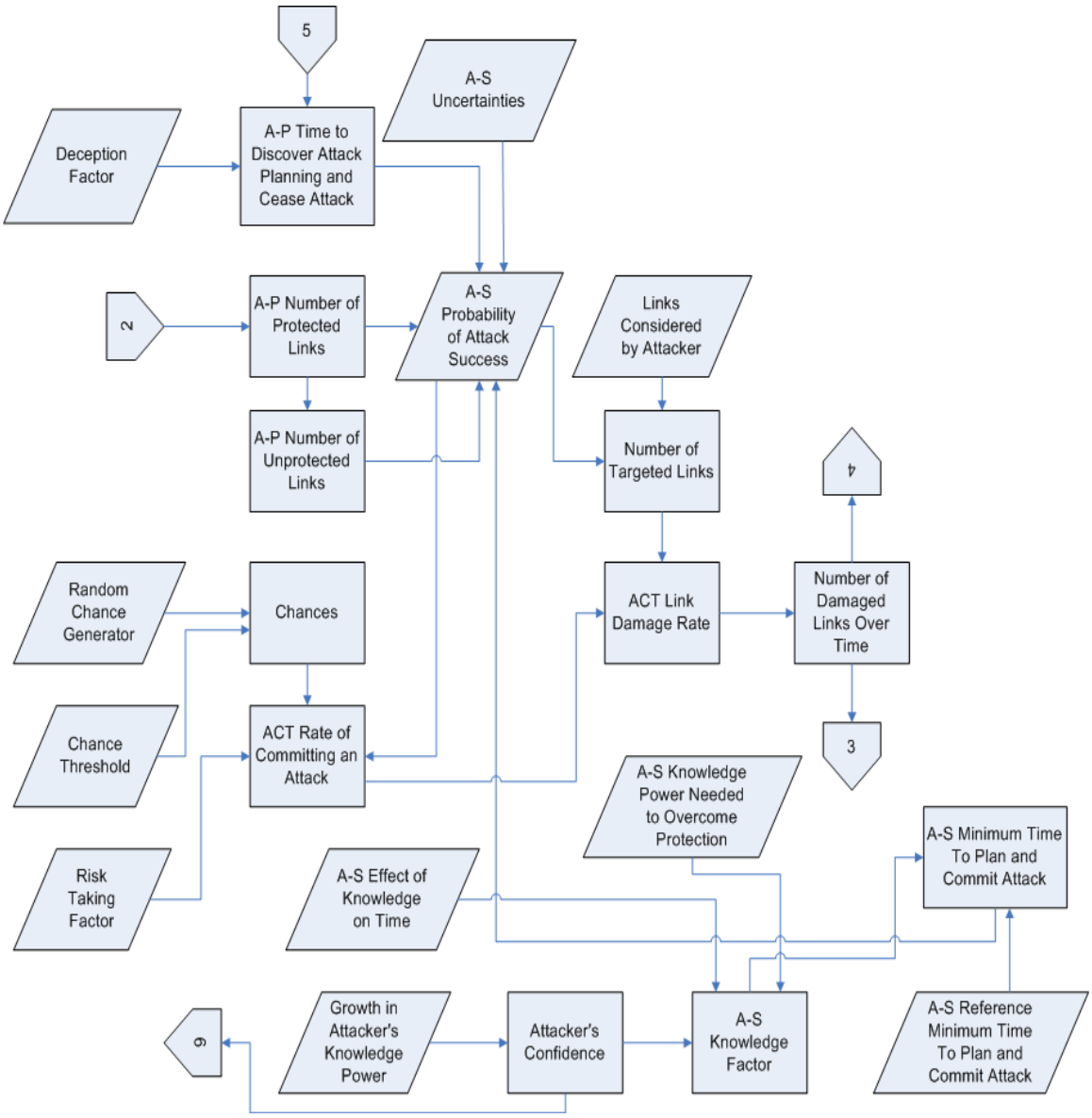


Figure 14: Attacker-Perception Model flow chart

by the attacker and how this data affects his decisions. The flow chart of this model is shown in Fig. 14.

Although it could be criticized that the P^2I^3-Model is designed to be used by the HS agency, using the information it has, and that this sub-model needs information that if known by the HS agency there would be no conflict at all, we clarify that the data used in

the attacker sub-model is data that would be derived from past experience with different attackers and is used to enable the evaluation of a protection deployment plan before it is actually implemented. More precisely, the attacker-related information in this model is in fact the scale that the HS agency uses to assess its proposed methodology of reducing the probability of attack success. As was the case with the HS-perception model, this sub-model has the "Growth in Attacker's Knowledge Power" as an input. Instead of having three rates; one for the HS agency, one for the attacker, and an actual rate, we made the second two as the joint variable mentioned above. This was done first for simplicity and second to represent a powerful attacker that has very good knowledge about the actual factors that control the growth rate of his knowledge. This rate, again, affects the attacker's confidence, then the A-S Knowledge Factor, the A-S Minimum time to Plan and Commit Attack, and finally the A-S probability of attack success. The A-S Probability of Attack Success is a function of the A-S Minimum time to Plan and Commit Attack, the A-S Uncertainties, Attacker Perceived (A-P) Time to Discover Attack Planning and Cease Attack, and the A-S Number of Protected and Unprotected Links.

It should be noticed that the attacker's perceptions, based on real-world factors, which were affected by the actions of the HS agency, helped him to estimate the probability that his attack will succeed. Next, the attacker makes a decision based on that probability of attack success, the number of links that he is considering for targeting, the available chances, and the inclination to take the risk. This decision determines the actual variable rate at which he will commit attacks which, in turn, determines the actual rate of link damage. Therefore, the decisions were converted to actions. The link damage rate

determines the actual number of damaged links over the study period. The number of damaged links affects the protection adjustment HS agency makes to the network and hence the number of protected links. The number of protected and unprotected links affect the A-P probability of attack success through the ”Protected Link Obfuscation Function” as will be seen in section 4.2.1.3. This means that the attacker’s actions affect his future behavior indirectly. Again, the variables and equations of this sub-model are summarized below.

- Growth in Attacker’s Knowledge Power, $G_{RK}(t)$: The main input to this sub-model. It represents the variables rate at which the attacker believes his information will grow in the period under study. It is measured in “BitsOfInfo/Year”. The values of this variable should be based on the experience information of the HS agency. These values are input as a rate value per year over the interaction period. Arbitrary variable rate was used in the simulations.

- Attacker’s Confidence, C_A: Attacker’s confidence is similar to $G_{RK}^{HS-A}(t)$ but here it is from the attacker’s point of view. It is measured in “BitsOfInfo” and is the integration of the rate of growth in attacker’s knowledge power. This is shown in Eq. 21.

$$C_A = \int_0^t G_{RK}(t)dt \tag{21}$$

- A-S Effect of Knowledge on Time, E_K^{A-S}: This is an input variable that represents the reduction effect that gaining information (by the attacker) has on the time needed to plan and commit attack from the attacker’s point of view. It is measured in “Years/BitsOfInfo”. The value of this variable is a constant the value of which is

determined from experiences of the HS with previous attackers and their perception of that effect.

- A-S Knowledge Power Needed to Overcome Protection, P_K^{A-S}: The amount of information the attacker believes is sufficient to overcome the deployed protection measured in "BitsOfInfo". It is an input constant.

- A-S Knowledge Factor, F_K^{A-S}: Represents the additional time added to the minimum ever time needed to plan and commit an attack if all necessary knowledge is available due to the difference in knowledge between the attacker and the required knowledge to overcome the protection. It is measured in "Years" and is calculated using Eq. 22.

$$F_K^{A-S} = (P_K^{A-S} - C_A) \times E_K^{A-S} \tag{22}$$

- A-S Reference Minimum Time To Plan and Commit Attack, $\hat{T}_{PCAr}^{A-S}$: An input variable similar to the one given in Eq. 6 that represents the time needed by the attacker, but this time in his perception, for preparing a plan and executing it if he has the desired information to overcome the deployed technology. Again, it is measured in "Years". Here, it is a constant that should be based on previous experiences of the HS agency with the considered attacker.

- A-S Minimum Time To Plan and Commit Attack, $\hat{T}_{PCA}^{A-S}$: The minimum time needed from attacker's perspective to plan for and commit an attack. It is equal to the minimum ever time that the attacker needs to prepare a plan and execute it – called reference minimum time - when all information is available. On the other hand, if the information needed to overcome protection is not available in full, this time becomes equal to that reference minimum time plus the time needed to acquire the rest of the information required (called A-S Knowledge Factor). This is given by Eq. 23

$$\hat{T}_{PCA}^{A-S} = \begin{cases} \hat{T}_{PCAr}^{A-S} + F_K^{A-S} & F_K^{A-S} > 0 \\ \hat{T}_{PCAr}^{A-S} & otherwise \end{cases} \quad (23)$$

- Deception Factor, F_D: An input constant that represents the multiplicative increase in the actual time needed to discover attack planning and to cease the attack, as perceived by the attacker. This increase is due to deception carried out by the HS agency to hide the actual time needed. The value of this constant should be based on intelligence efforts of the HS agency. It is a dimensionless variable.

- A-P Time to Discover Attack Planning and Cease Attack, T_{DPCA}^{A-P}: Attacker's perception of the minimum time needed by the HS agency to discover his planning for an attack and to cease that attack. This time is measured in "Years" and is the multiplication of the deception factor and the actual time needed by the HS agency T_{DPCA}^{ACT} as shown in Eq. 24.

$$T_{DPCA}^{A-P} = T_{DPCA}^{ACT} \times F_D \quad (24)$$

- A-S Uncertainties, U^{A-S}: The input representing the effect of lack of information on the attacker's side on the probability of his success in the attack P_s^{A-S}. This input constant is "Dimensionless" as should depend on HS's past experiences. It takes values between 0 and one, where 0 means full certainty and 1 represents complete uncertainty about the acquired information.

- A-P Number of Protected Links, N_P^{A-P}: Represents the perceived number of protected links by the attacker. It was made equal to the link obfuscation function's output in this model. As mentioned before, the link obfuscation function is determined by the HS agency and is intended to try to address the perceived protection by the attacker.

So, it is not the same as the actual perception of the attacker but has been set equal to the A-P number of protected links for model simplicity.

- A-P Number of Unprotected Links, N_{UP}^{A-P}: Clearly, this is equal to the rest of the links as shown in Eq. 25.

$$N_{UP}^{A-P} = N_L - N_P^{A-P} \tag{25}$$

- A-S Probability of Attack Success, P_s^{A-S}: This variable expresses the probability that an attack will be successful for the attacker's point of view and based on his perception/collected information. It is a function of the A-S Minimum time to Plan and Commit Attack, the A-S Uncertainties, A-S Time to Discover Attack Planning and Cease Attack, and the A-S Number of Protected and Unprotected Links. Eq. 26 represents the probability as a function of these variables. It is clear that as the ratio of the number of the A-S number of unprotected links to the total number of links become larger, the probability of success increases. When the time to discover attack planning and cease the attack becomes larger than the time needed to plan and commit attack, the probability of attack success increases. Finally, when the uncertainty increases, this probability decreases.

$$P_s^{A-S} = \frac{N_{UP}^{A-P} \times T_{DPCA}^{A-P} \times (1 - U^{A-S})}{(N_{UP}^{A-P} + N_P^{A-P}) \times (T_{DPCA}^{A-P} + \hat{T}_{PCA}^{A-S})} \tag{26}$$

- Chances, C: This variable is equal the value of a uniform random variable that takes values from 0 to 100 if that value is greater than a pre-specified threshold, and is zero otherwise. The variable represents the chances exploited by the attacker. This variable is measured by the number of chances per year i.e. the unit is "1/Year". Eq. 27 shows this variable.

$$C = \begin{cases} X_C & X_C > T_C \\ 0 & otherwise \end{cases} \tag{27}$$

- Chance Threshold, T_C: An input constant that specifies the strength of a chance that makes the attacker willing to take it. This should be based on HS agency's with different attackers and the attractiveness of different chances to them. It takes values between 0 and 100 as well.
- Risk Taking Factor, F_R: A dimensionless input constant the value of which determines the inclination of the attacker to take the risk and commit an attack even when there is a possibility that this attack will fail. A scale from 0 to 10 for example could be used to represent different levels of inclination. A value of 3 was used in the simulation.
- Number of Targeted Links, N_{tL}: An input representing the variable number of links considered for targeting over time. As was the case with the link obfuscation function, an arbitrary function was used to model this variable. In this case, it is a step function that changes over time. The determination of this input is based on intelligence information about system elements - links here - that represent high value for the attacker.
- ACT Rate of Committing an Attack, R_{CA}^{ACT}: The actual rate at which the attacker commits an attack is governed by his perception of the probability of attack success, by the chances he is willing to take, and by the inclination to take a risk even when success is not guaranteed. This is shown in Eq. 28. This rate is measured in number of attacks per year, i.e. "1/Year".

$$R_{CA}^{ACT} = P_s^{A-S} \times C \times F_R \times N_{tL} \tag{28}$$

- ACT Link Damage Rate, R_{LD}^{ACT}: The rate at which the links are actually damaged. It is governed by the actual probability of attack success. The probability specifies a fraction

of the rate at which the attacker tries to commit attacks to be the actual rate of damaging links. This rate is measured in "Links/Year" and is given by Eq. 29. This variable is more a mixed variable than an attacker-model variable as it is a common variable between this model and the real-world model.

$$R_{LD}^{ACT} = R_{CA}^{ACT} \times P_s^{ACT} \tag{29}$$

- Number of Damaged Links Over Time, N_{DL}: The number of damaged links over time is determined by the actual link damage rate. The former is a level variable and is therefore the integral of that rate over time. Eq.30 shows this relation.

$$N_{DL} = \int_0^t R_{LD}^{ACT} dt \tag{30}$$

c. Real-World Model:

The third and last model is concerned with the representation of real world, and what is actually happening away from the perceptions of the opponents. Again, we restate that although the variables in this sub-model will not likely be known by any of the two sides of the interaction, the HS agency will be using values that are based on experience to serve as a firmer reference than that of the attacker to assess its gained information and its methodology. The flow chart of the RW model is shown in Fig. 15. Most of the variables in this model are similar in the way they function to their counterparts in the two previous models. The ACT probability of attack success is affected by many factors: the number of protected and unprotected links, the actual time to discover attack planning and cease attack (this time increases exponentially over a long period (>>10 years) of time in this model to account for the time evolving opposing conditions that may hamper HS

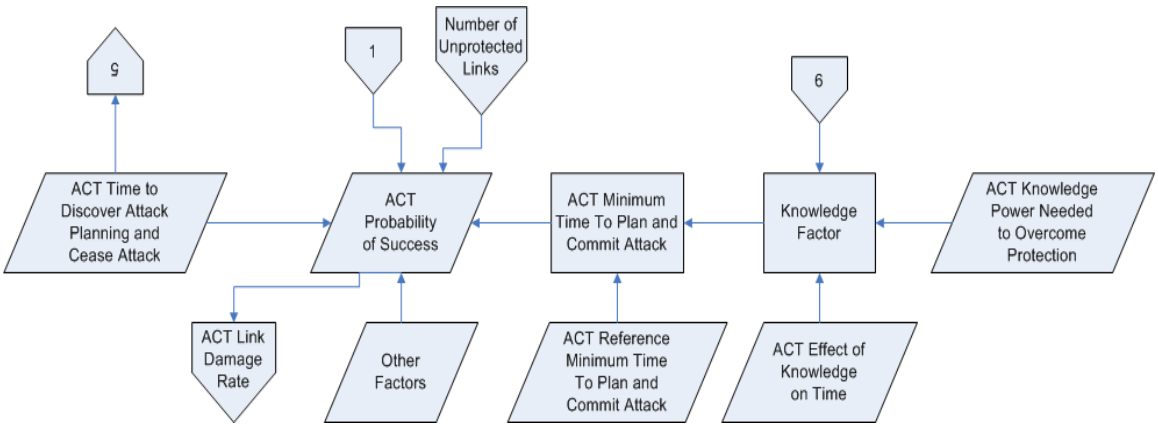


Figure 15: RW Model flow chart

agency's efforts to discover the attack), the actual minimum time to plan and commit attack, and on 'other factors' (these are factors that could be added in the future to refine the model. In the current model they are represented as a constant equal to 0.1. The complement '0.9' of this constant is multiplied by the remaining terms that affect the actual probability). A summary of the real-world model variables and equations is given below.

- ACT Knowledge Power Needed to Overcome Protection, P_K^{ACT}: A constant similar to the knowledge power of the other two sub-models, but in this case represents the actual or the correct level needed to overcome the protection. It is again measured in "BitsOfInfo".
- ACT Effect of Knowledge on Time, E_K^{ACT}: An input constant expressing the actual reduction effect "Years/BitsOfInfo" on the time needed to overcome protection as a result of gaining information on the attacker's side.
- Knowledge Factor, F_K: The reduction in the time needed to overcome protection as a result of gaining information. Eq. 31 describes this variable.

$$F_K = (P_K^{ACT} - C_A) \times E_K^{ACT} \tag{31}$$

- ACT Minimum Time To Plan and Commit Attack, $\hat{T}_{PCA}^{ACT}$: The 'actual' version of the similar variable in the attacker-model. It is a function of the actual reference minimum time to plan and commit attack and the knowledge factor as was the case with the A-S minimum time to plan and commit attack. Eq. 32 follows is very similar to Eq. 23.

$$\hat{T}_{PCA}^{ACT} = \begin{cases} \hat{T}_{PCAr}^{ACT} + F_K & F_K > 0 \\ \hat{T}_{PCAr}^{ACT} & otherwise \end{cases} \tag{32}$$

- ACT Probability of Success, P_s^{ACT} : A real-world model output variable that models the actual probability of attack success. It is a function of the actual number of protected and unprotected links, the actual times for planning and committing an attack and for discovering that attack and ceasing it, and finally, on other factors. These other factors represent any non-modeled contributors to the actual success probability. They were included to enable adding more terms to the probability in the future to refine the model. The variable 'other factors' takes values from 0 to 0.995. The lower limit indicates no effect for these factors on the probability, while the upper limit indicates a very strong effect on the probability. The actual probability of attack success is given by Eq. 33.

$$P_s^{ACT} = \frac{N_{UP}}{(N_{UP} + N_P)} \times \frac{T_{DPCA}^{ACT}}{(T_{DPCA}^{ACT} + \hat{T}_{PCA}^{ACT})} \times (1 - F_O) \tag{33}$$

4.2.1.3. Inter-Model Relationships:

The interdependencies between the three sub-models are discussed in this sub-section. Three main inter-model relationships are of interest: a relationship through the

number of protected and unprotected links, another relationship through the protected link obfuscation function, and a third through actual link damage rate.

a. Number of Links and Protected/Unprotected Links:

The number of links and the number of protected/unprotected links affect the HS-S probability of attack success and the ACT probability of attack success directly. They also affect the A-S probability of attack success indirectly through the so called link obfuscation function. When the HS agency makes a protection adjustment to the network, this is reflected in the three probabilities. Therefore, the perception of the HS agency generates actions that make a real-world change that in turn affects the perception of the attacker, and changes the actual situation. On the other hand, the affected perception of the attacker imposes a change on his actions, which also affects the real-world. This gives an insight into the nature of the interaction between the two entities in the surrounding frame.

b. Link Obfuscation Function:

The “Protected Link Obfuscation Function” is an HS-perception model variable. It is a mathematical representation of the effect of installing illusive devices, e.g. empty black boxes, to mislead the attacker and make protected and unprotected links indistinguishable. This function operates on protected and unprotected links, and affects the perception of the attacker about the number of protected links. This, as a result, contributes to the formation of the perception of the probability of attack success from his perspective. It is clear again how an action done by the HS agency has an effect on the perception of the attacker, and hence actions, through the real world.

Finding a suitable mathematical formulation for the effect of this deployed obfuscation on attacker's perception is one difficulty that may require a considerable effort from HS. In this model, we used an arbitrary function that acts as an operator on the number of protected and unprotected links and produces misleading numbers of these links while not being too far from actual values to maintain realism.

c. Link Damage Rate:

The ACT Link Damage Rate depends directly on the ACT Probability of Attack Success, and in- directly on the A-S Probability of Attack Success. Also, the ACT Probability of Attack Success depends on the number of protected and unprotected links which are the result of the efforts of the HS agency.

4.2.1.4. Model Critique:

The P^2I^3−Model can effectively describe the interactions between the homeland security agency responsible for protecting the system under consideration and the attacker. It also provides the agency with a deployment road-map and triggers the decision correction process. There are, however, some considerations that, if incorporated in the model, would make it more powerful and more realistic. For example, in the base model, the deployment road-map only provides a time frame for protecting links and the necessary levels of that deployment at each point, however, it does not provide an order in which these links should be protected nor includes link importance ranking in the deployment decision process. Including such ranks and order of deployment would be very helpful in increasing network vulnerability and would definitely affect the inclination of the attacker to take the risk and the probabilities of attack success. Also, the

base model dealt only with one attacker type which is not realistic. Having multiple attacker types each with their own set of variables should serve the realism of the model. Considering different network elements instead of only links and considering the non-attack alternative are two other important considerations (Jenelius et al.). Finally, the baseline model did not deal with links as actual entities; instead, it dealt with the number of links. More precisely, when the attacker considers some links for targeting, the model uses only the number of such links without specifying them individually. This is not accurate as it gives attacking any set of links the same importance to the attacker which is unrealistic. Therefore, the model should deal with links as separate entities and with link numbers.

4.2.1.5. Model Refinement:

In this section we focus on refining the model by considering link ranks. The model was originally developed using VenSim™. In order to have more freedom in extending the model and be able to integrate it with other existing models, we wrote our Java™ version of the P^2I^3−Model.

Protection deployment is not an even process where the available technology is just deployed in a random order and with the same technological level for all system elements. It should follow a well-studied methodology to achieve the desired goal. There are two perspectives from which this problem should be tackled. The first is the order of link protection and the second is the technology grade used for each system level. We consider the former only in this work. To incorporate link ranks; which represent the level of link importance to the attacker, in the baseline model, the output of the previously discussed (see Chapter II) vulnerability assessment model (Wang et al. 2011)

is used as an input to this model with a slight modification. The original output is a list of link failure probabilities. The failure probability of each link was mapped to a corresponding 'link weight'. Links with higher probabilities of failure get higher weights. This link-weight list was used along with the deployment road-map resulting from the baseline model to provide a more specific deployment plan to the HS agency. In addition, these weights were used to model the effect of protecting the highly vulnerable links first on the probabilities of attack success for each of the three sub models. A new term that stresses the effect of link rank/weight on the probability of attack success was added. Eqs. 34, 35, and 36 show the three probabilities with the effect of link ranks included:

$$P_s^{HS-S} = \frac{N_{UP} \times W_{UP} \times U^{HS-S}}{(N_P + N_{UP}) \times (W_{UP} + W_P) \times O_e} \tag{34}$$

$$P_s^{A-S} = \frac{N_{UP}^{A-P} \times W_{UP} \times T_{DPCA}^{A-P} \times (1 - U^{A-S})}{(N_{UP}^{A-P} + N_P^{A-P}) \times (W_{UP} + W_P) \times (T_{DPCA}^{A-P} + \hat{T}_{PCA}^{A-S})} \tag{35}$$

$$P_s^{ACT} = \frac{N_{UP} \times W_{UP}}{(N_{UP} + N_P) \times (W_{UP} + W_P)} \times \frac{T_{DPCA}^{ACT}}{(T_{DPCA}^{ACT} + \hat{T}_{PCA}^{ACT})} \times (1 - F_O) \tag{36}$$

Where:

W_{UP} : Sum of the weights of the currently unprotected links.

W_P : Sum of the weights of the currently protected links.

The mapping process from link failure probabilities is simple. The idea is to put more emphasis on links with higher failure probabilities. The links are first sorted in a descending order of their failure probabilities. Then they are ranked, from one to the number of links, based on this sorted order. The simple formula given in Eq. 37 is then used to find the corresponding link weights.

$$W_i = 100 \times \frac{P_i^f}{R_i} \tag{37}$$

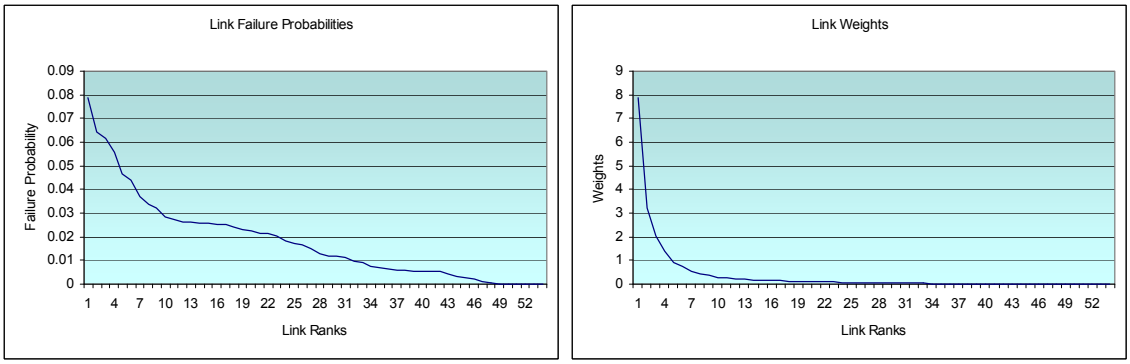
Where:

W_i: Weight of link i

P_i^f: Failure probability of link i

R_i: Rank of link i

An example of this ranking process and the resulting emphasis on links with higher failure probabilities is shown in Fig. 16, where a set of 55 links were used as a sample. It is clear from the figure that, as expected, for the highly ranked links, i.e. links with high failure probability, the corresponding weights are very high, while for lower rank links, the weights decrease rapidly. This has the effect of emphasizing highly ranked links as mentioned above.



(a) (b)

Figure 16: Sample link weighing process. a) Link failure probabilities, b) Corresponding link weights

4.2.2. Sensor Deployment

The problem of sensor deployment is application specific. A deployment valid for environmental monitoring or structure robustness reporting is not suitable for a target tracking or a traffic monitoring application. The problem under consideration: traffic network protection, is a sensitive problem where any delay or lack of information could cause a disaster or mass loss of lives. The specific context of monitoring explosive devices which has been selected requires that some specific links of high importance get more attention than other less important ones. It also requires that the protection of these links not be only present at the links themselves or their vicinity, but to extend up to a suitable distance that ensures that the attacker is detected and ceased far enough from the place of interest. Due to the above requirements, and due to other WSN design considerations such as network longevity or lifetime extension as well as 'hole' prevention, and network reliability, we chose the proposed deployment techniques for transportation network links. In the following sections, we will discuss the two proposed deployment methods in details, and provide the rationale for choosing each of them WSN-wise and security-wise. It is necessary, however, to first introduce computational geometry (CG) and some of its algorithms and concepts to aid the understating of the proposed protocols.

4.2.2.1. Computational Geometry:

Computational geometry is the branch of computational science responsible for dealing with different geometry aspects such as: 1) the representation of geometric primitives like points, lines, polygons, curves, etc., 2) the associated algorithms, 3) geometry relationships such as geometry intersections, unions, distances, etc., and 4)

geometry operations like: finding the centroid of a shape, the convex hull of a group of points, Delaunay triangulation, Voronoi diagrams, shape simplification, etc. In the next set of subsections, we describe some of the CG algorithms that we exploit in our sensor deployment algorithms.

4.2.2.1.1. Skeletons:

One can think of a skeleton in the exact way a human skeleton can be though of. This is one of the reasons it is called a skeleton. Skeletons describe the main structure of a shape. In the same way a human skeleton gives a very good idea of how a human looks, skeletons of geometric shapes provide information about the general features of a shape. On the other hand, much like the skeletons of human beings do not show the exact details of how their facial features are, for example, shape skeletons do not necessary describe the minute local information of a shape (although some implementations can do so); instead they describe the global information. To illustrate the idea of skeletons, we borrow the prairie example from Gonzalez and Woods (2008): Imagine that a fire catches the boundaries of a uniform, dry grass field; the fire fronts will start moving inwards at the same speed. The set of points where the fire fronts meet 'reach at the same time' represents the skeleton of the field's shape. The formal definition of a skeleton as provided by Gonzalez and Woods is based on the definition of the so called medial axis transform (MAT) proposed by Blum (1967) as follows:

"The MAT of a region *R* with border *B* is as follows: For each point *p* in *R*, we find its closest neighbor in *B*. If *p* has more than one such neighbor, it is said to belong to the medial axis (skeleton) of *R*"

In the above definition, closeness is determined by the definition of distance. In this work we deal with Euclidean distances only.

It is worth to mention that there are many algorithms that were developed to obtain the skeleton of a shape. We refer the reader to Ref. (Gonzalez and Woods, 2008) for more information about these algorithms. Fig. 17 provides some examples of shape skeletons generated using the ImageJ free image processing and analysis tool[1].

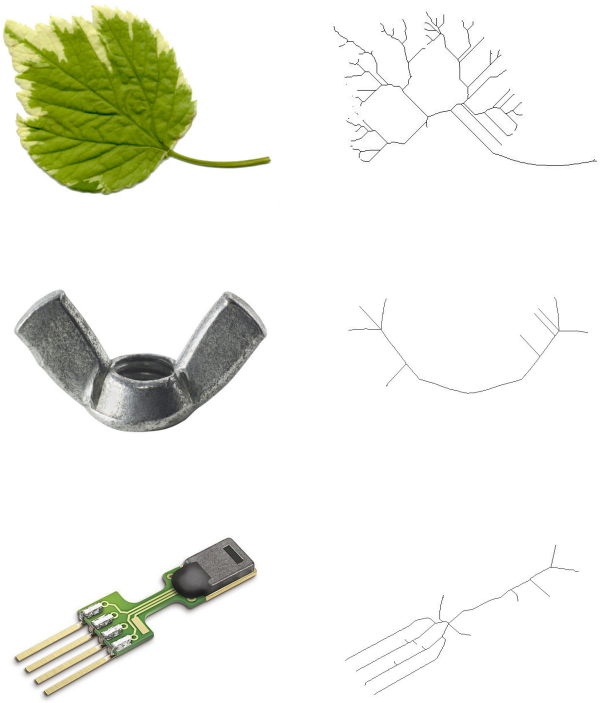


Figure 17: Examples of shape skeletons.[2]

4.2.2.1.2. Convex Hull:

Another very important concept in CG is the convex hull. For a set of points, one can think of a convex hull as a rubber band that tightly encloses this set of points. In order to provide a formal definition of the convex hull, two basic concepts should be clarified

[1] ImageJ; Image Processing and Analysis in Java: http://rsbweb.nih.gov/ij/
[2] Leaf: http://rsbweb.nih.gov/ij/; wing nut: http://photo-dictionary.com/; humidity sensor: http://www.directindustry.com/

first; convex set and convex combination (Boyd and Vandenberghe (2004)). A convex set is a set of points containing all the line segments (and their points of course) connecting any two points of the set. This is expressed mathematically as:

$$\theta x_1 + (1-\theta)x_2 \in C \qquad \forall\ x_1, x_2 \in C \quad ,0 \le \theta \le 1 \tag{38}$$

Figs. 18 (a) and (b) show an example of a convex and a non-convex set. A convex combination on the other hand is defined as follows:

Given a set of points $x_1, x_2, ..., x_n$ any point of the form:

$$x = \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n, \quad \theta_1 + \theta_2 + ... + \theta_n = 1, \quad \theta_i \ge 0 \tag{39}$$

represents a convex combination of the n points.

The two above definitions lead to the definition of a convex hull. A convex hall of a set of points is simply the set of all convex combinations of the points in that set. Figs. 18 (c) and (d) show examples of the convex hulls of two point sets.

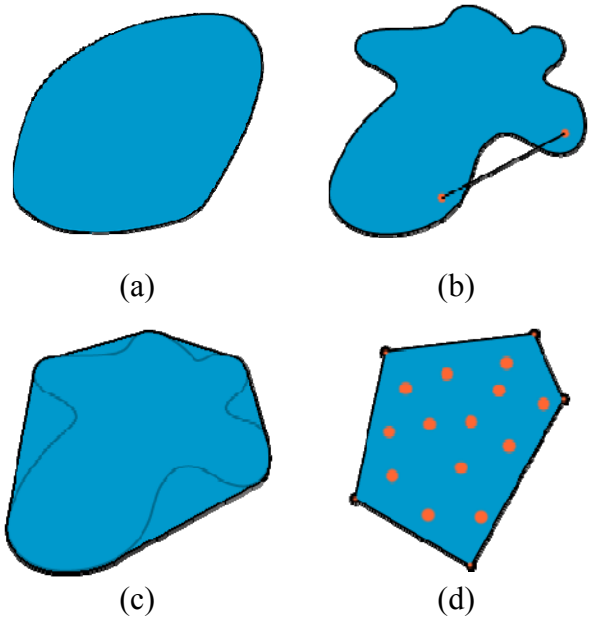


Figure 18: Convex hull concept (a) Convex point set (b) Non-convex point set (c) and (d) Convex hull examples

4.2.2.1.3. Voronoi Diagrams and Delaunay Triangulation:

Two related and very important CG concepts are Voronoi diagrams and Delaunay triangulation. Starting with Voronoi diagrams, the problem is defined as follows (Miu, 2001): Given a set S of n distinct points p_i ($p_i \in n$, $1 \le i \le n$) in a plane, find a set of n cells, one per point (also called site) that divide the plane with the condition that a point q is considered to lie in the cell associated with site p_i *iff* the Euclidean distance from that point to p_i is less than the distance to any other site $p_j \in S$. An example of a Voronoi diagram in shown in Fig. 19; please note that this diagram is a part of a bigger diagram and therefore the boundaries are not descriptive of a typical boundary of a Voronoi diagram. More clearly, a typical Voronoi diagram would have what is called 'unbounded cells' at the boundary; i.e. edges at the boundary would be extending to infinity forming half planes. On the other hand, the cells that are closed are called 'bounded cells' and lie in the middle of the diagram. An edge in Voronoi's diagram is a subset of the locus of points that are equidistance from two sites p_i and p_j. Vertices are, informally, the meeting points of these edges. The time complexity of finding a Voronoi diagram for a set of n points in 2D is $O(n^2)$ if brute force is used. On the other hand, a complexity of $O(n \log n)$ can be achieved using Fortune's algorithm (Fortune, 1986).

Delaunay triangulation is the dual of Voronoi diagram. The problem is defined as triangulating a set of n points in a plane such that no point is in the circumcircle of any other triangle (Sedgewick and Wayne (2007)). As Delaunay triangulation is the dual of a Voronoi diagram for a set of points, Furtune's algorithm can also be used to find Delaunay's triangulation. Therefore, the time complexity of Delaunay's

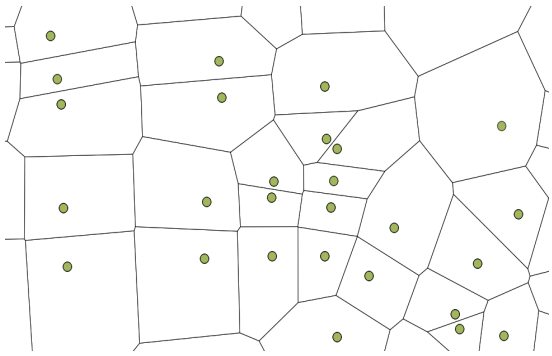
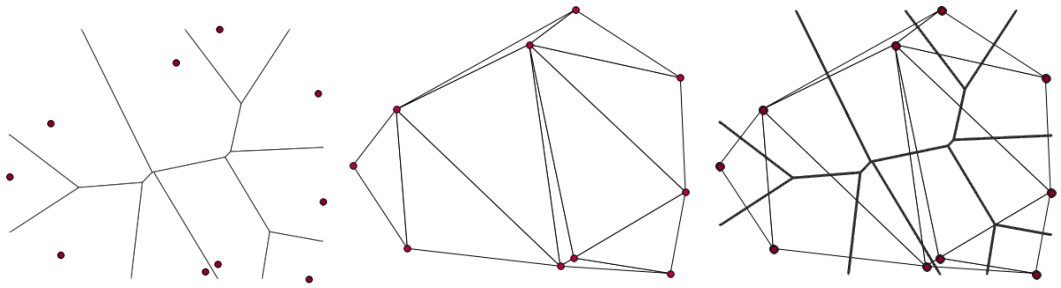


Figure 19: An example of Voronoi Diagram (generate using Quantum GIS©)



(a) (b) (c)

Figure 20: An example of: (a) Voronoi Diagram (b) Delaunay Triangulation (c) Voronoi and Delaunay (generate using Quantum GIS©)

triangulation is $O(n \log n)$ as well. If brute force is used instead, the time complexity becomes $O(n^4)$ as stated by Sedgewick and Wayne. Fig. 20 (a) and (b) show examples of Voronoi and Delaunay, and (c) shows both Voronoi and Delaunay for the same set of points. It is also worth to mention that boundary of the Delaunay triangulation is the convex hull.

4.2.2.1.4. Centroid:

The centroid of a planar shape is the intersection of all straight lines that divide the shape into two parts of equal moment about the line, Wikipedia (2011). It can be simply thought of as the arithmetic mean of all points of the shape. There are many algorithms used to obtain the centroid of a shape. One algorithm depends on the triangulation discussed above as follows. First, a triangulation of the shape is done using the Delaunay triangulation. Then, the centroids of the resulting triangles are found by taking the average of the three vertices of each triangle. Finally, a sum of the triangles' centroids weighted by the corresponding triangles' areas is found, and is normalized by the total area of the shape[11]. This is only one example and is not necessarily the best available. Fig. 21 shows some examples of shape centroids generated using Quantum GIS©.

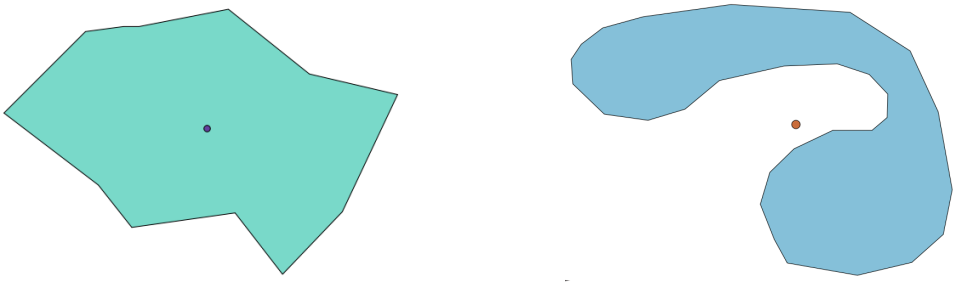


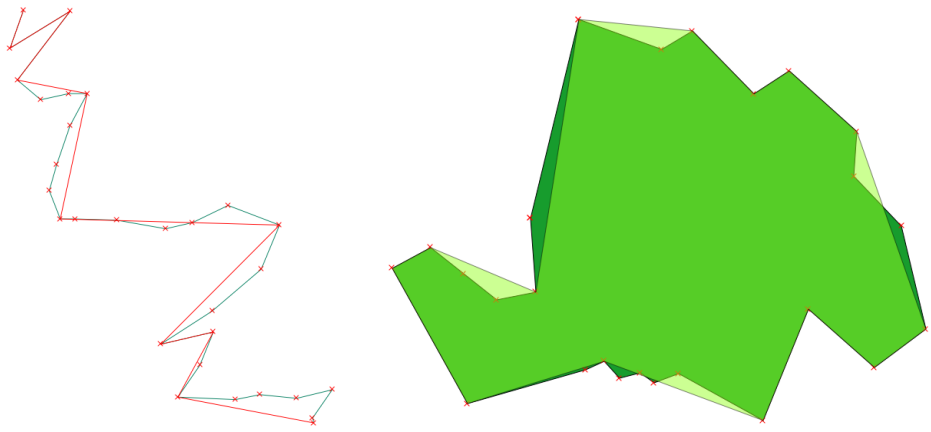
Figure 21: Examples of polygon centroids

4.2.2.1.5. Douglas-Peucker's Simplifier

Sometimes it is necessary to simplify a curve approximated by a set of points (also called a poly-line) to get a new poly-line with a fewer number of points subject to a

[11] Finding the centroid of a polygon, Subject 2.02, online: http://www.faqs.org/faqs/graphics/algorithms-faq/

certain tolerance. This is what a Douglas-Peucker's algorithm does; Douglas and Peucker (1973). Given a set of n points connected in a specific sequence to form a poly-line as shown by the longer line in Fig. 22 (a), the objective is to find a subset of the points that simplifies the given poly-line while not loosing the main features, i.e. within a certain tolerance (describing the allowed distance to the original curve). The algorithm is recursive and works by marking the start and end points to keep them and then finding the point with the maximum perpendicular distance to the line connecting these two points from among all other points. If that distance is greater than the tolerance, the point is kept and two recursive calls to the algorithm are made passing the start point and the newly kept point to the first call, and the newly kept point and the end point to the second call. Otherwise, all other points that were not chosen to be kept yet can be discarded. At the end, the kept points represent the new points. The shorter poly-line in Fig. 22 (b) represents the simplified curve for a tolerance of 0.1. Fig. 8 (b) is an example of a



(a) (b)

Figure 22: Examples of (a) A Douglas-Peucker-simplified poly-line.

(b) A simplified polygon.

simplified polygon using Douglas-Peucker's algorithm. The complexity of this algorithm is $\theta(n \log n)$ and in the worst case: $O(n^2)$.

4.2.2.2. Deployment Algorithm:

The development of a deployment algorithm suitable for different transportation networks and different scales is not trivial. This suggests the proposal of different deployment schemes and making comparisons between them to arrive to the best possible deployment. We propose two deployment algorithms for this purpose; each has its rationale. It remains necessary, however, to study the cost and performance aspects before the proposed deployments are actually implemented. We also believe that separating the design of the deployment algorithm and the WSN packet routing protocol is impractical. They should be designed simultaneously to guarantee the best results, as the performance does not depend only on the deployment algorithm, but also on the routing protocol. For this reason we kept in mind several routing factors while designing the deployment protocols.

4.2.2.2.1. Method 1

In GIS, geographic data is represented as features that are usually stored in some specific format. A feature could be a street, city border, river, police station, park, sensor, etc. One very common format for storing GIS data is the so called shapefile (more precisely: Esri[12] shapefile). A shapefile stores geometric primitives, such as points, lines, polygons, etc., that represent features. In both of our algorithms, the input is a shape file containing GIS data about the different transportation links in the network under

[12] http://www.esri.com/

consideration. Links are usually stored as line-strings or multi-line-strings which are simply groups of small line segments connecting points to form lines and poly-lines that closely resemble curves. There are two output formats for both algorithms; the first is a shapefile storing the deployments' sensor locations as point coordinates (for display purposes only), while the second provides deployment information in textual format. The second output is the one intended for use by the deployment personnel. We selected Chicago's road network and Delaware's road system as our test networks. Therefore, we use them from here on.

The high-level description of the idea of the first algorithm is as follows: First, an envelope that very closely encloses all network links is found. Then the skeleton of that envelope is found. This will provide a descriptor for the overall shape of the network. After that, we get all the terminal and junction points of the skeleton (as shown in Fig.23) and find the combinations of all three points of this point set. For each combination a triangle is formed. Then, for all the remaining terminal and junction points of the skeleton, the distances to that formed triangle are found. The maximum distance is then compared to a tolerance (a specific predefined length), and if the maximum distance is shorter than that tolerance, the triangle is chosen to be used in the next step. Otherwise, the combinations of all four points are found and the process repeats. If this fails again, the next set polygons with an additional vertex are found, and the comparison takes place. For the network under consideration, the process ended with a four vertex polygon as shown in Fig. 24. The rationale for doing this step is to find the polygon with the smallest number of vertices that can cover most of the network's area. These vertices are used

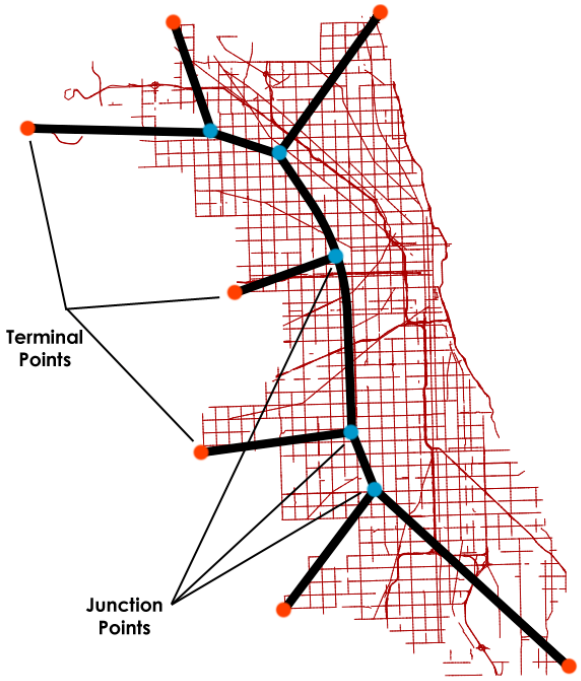
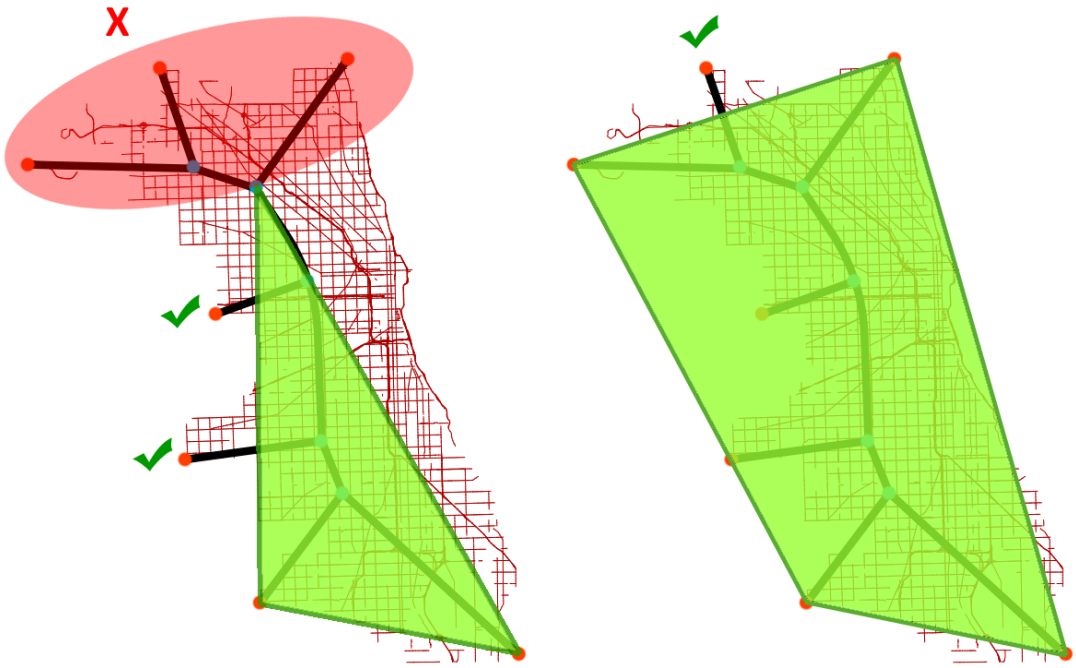


Figure 23: Skeleton with terminal and junction points defined

afterwards for placing the base stations (BSs). By selecting the locations of the base stations at or near the corners of the network, deployment scalability is assisted. If a neighboring city decides to deploy sensors as well, the BSs of the deployed WSN can serve the adjacent parts of that city by using Omni-directional antennas instead of directional antennas. This is shown in Fig. 25. The next step is to find the centroid of the envelope; this will be the position of placing the central Command Center (CC) responsible for collecting the data from the WSN and for issuing queries or notifying emergency personnel. Placing the BSs at the corners and the CC at the center reduces the data traffic congestion at the nodes surrounding the CC, prevents them from dying quickly, and extends the network lifetime as a whole. Up till now, what has been done is that the places of the BSs and the CC were determined. The next step is to find the locations of sensors themselves. As the deterministic deployment in the case of a large

scale deployment would be impractical due to the huge number of sensors used, our approach is a mixture of deterministic and random deployment. More precisely, we can say that it is almost random; the main component is that sensors are randomly deployed. The smaller deterministic component is responsible for assuring that sensors have higher concentrations as they get closer to the BSs and the CC and lower ones radially outwards from their locations. This choice was made to help mitigate the quick node death problem near the BSs and the CC (and the resulting hole problems) from a deployment perspective. Because sensor nodes from all around the network collect and send data towards the BSs and the CC based on closeness, the traffic near them is high most of the time and energy depletion becomes fast. This causes these nodes to die faster than other



(a) (b)

Figure 24: Finding the polygon with minimum number of vertices that covers most of the network links. (a) A triangle is not sufficient. (b) A four vertex polygon is sufficient.

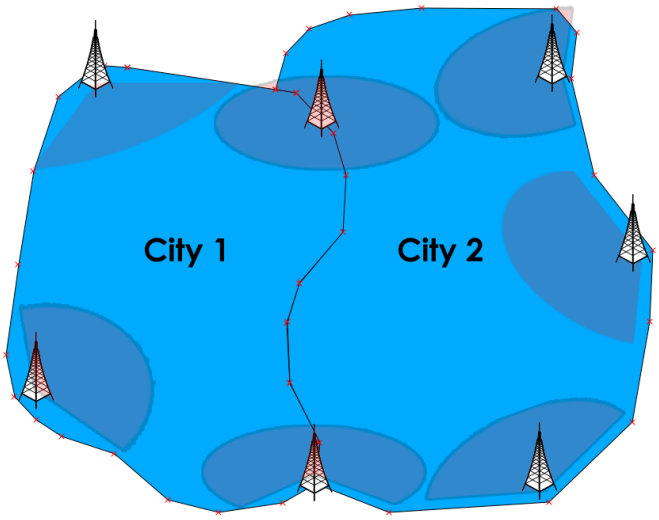


Figure 25: Scalability of the BS deployment

nodes, leading to connectivity loss and the so-called 'hole problem' (A hole exists in the network when a node finds no closer node to the BS than itself, and the same time, it cannot transmit the packet directly due to long distance or power limitations). For this purpose, we select a number of deployment radii exponentially increasing in value as they move away from the BSs and the CC. The number of such radii is a design parameter. On the other hand, the sensor density decreases exponentially moving away from the BSs. Again, the parameters of this exponential function are design parameters. Eqs. 40 and 41 are used to determine the deployment radii and the number of sensors per inter-radius ring respectively. Eq. 42 determines the deployment start radius. Fig. 26 shows the generated radii and the resulting sensor distribution for Chicago's network. In addition, Fig. 27 shows a sample output containing the deployment information.

$$R_i^D = R_s^D \times \exp(F^S i) \qquad ,0 \le i < N_R \tag{40}$$

$$N_i = \mathrm{C} \times \exp(-R_i^D / R_m^D) \tag{41}$$

$$R_s^D = 0.5 \times R^T \tag{42}$$

Where:

F^S : Deployment smoothness factor (smoothness of the sensor gradient)

R_i^D : Deployment radius i

R_s^D : Deployment start radius

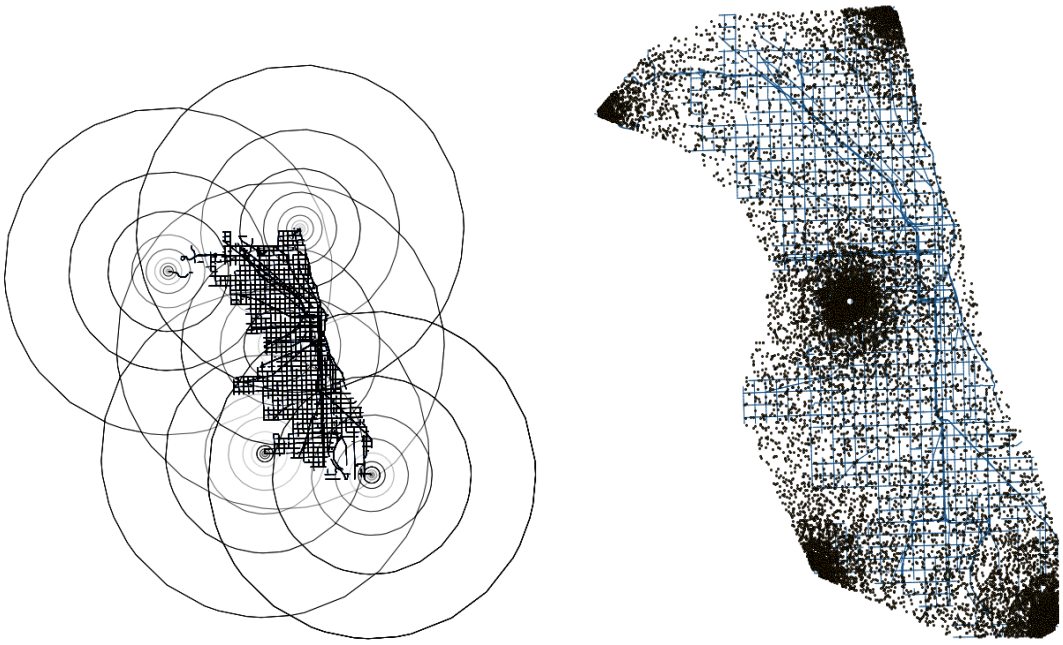
N_i : Number of sensors within radius i

R_m^D : Maximum deployment radius

C : Constant (was set to 1000 in the tests made)

R^T : Base station / command center transmission range

N_R : Number of deployment radii



(a) (b)

Figure 26: Algorithm 1 (a) Deployment Radii (b) Sensor distribution

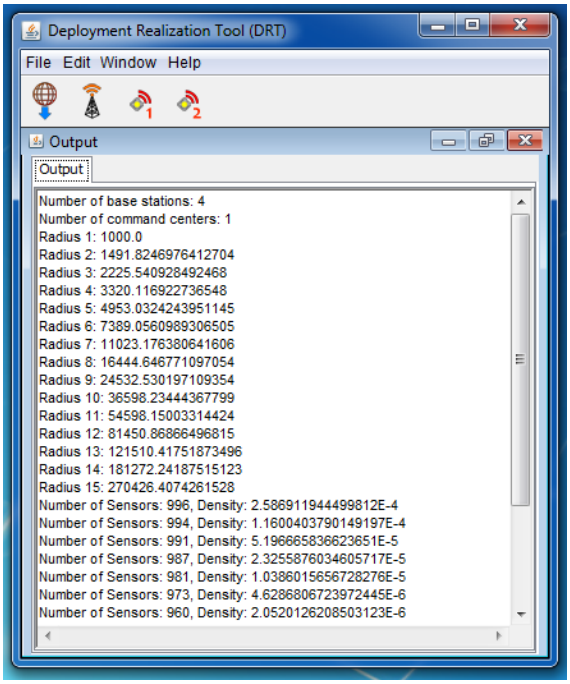


Figure 27: Deployment method I: sample output

One detail not discussed yet is the determination of the tight envelope that closely encloses the network links. First, points constituting the network links are extracted, i.e. we deal with a point cloud in 2D. Next, Delaunay's triangulation is used to triangulate this set of points. If the resulting triangles are combined, we get the convex hull, but this not what we need. Therefore, to overcome this problem, we do two tests on each triangle:

Test 1-1: $A_i^T > const. \times A_{Max}^T ?$ $\quad ,1 \le i \le M$

Test 1-2: Does A_i^T contain any points from the point set?

Test 2: $L_i^T < const. \times L_{Max}^T ?$ $\quad ,1 \le i \le M$

Where:

M : Number of triangles

A_i^T : Area of triangle i

A_{Max}^{T} : Maximum-area triangle

L_i^T : Perimeter length of triangle i

L_{Max}^{T} : Maximum- perimeter-length triangle

const. : Taken to be 0.4 (design parameter)

If the answer to the two parts of the first test is yes, the triangle is stored. If any of them fails, test 2 takes place and the triangle is stored in the case of success. If none of the tests succeeds, the triangle is discarded. Eventually, the stored triangles are combined to form the envelope. This process is given by Algorithm 1-a shown in Table 2. The purpose of the first test is to get rid of triangles with large areas that do not contain any points, i.e. any links. The second test takes care of the triangles with small areas but with long edges. These triangles are usually on the boundary and connect far points and should not be kept. Two examples of triangle removal are given in Figs. 28 and 29. The left part of each figure shows the original triangles and the envelope after triangle removal, while the right shows the network as well.

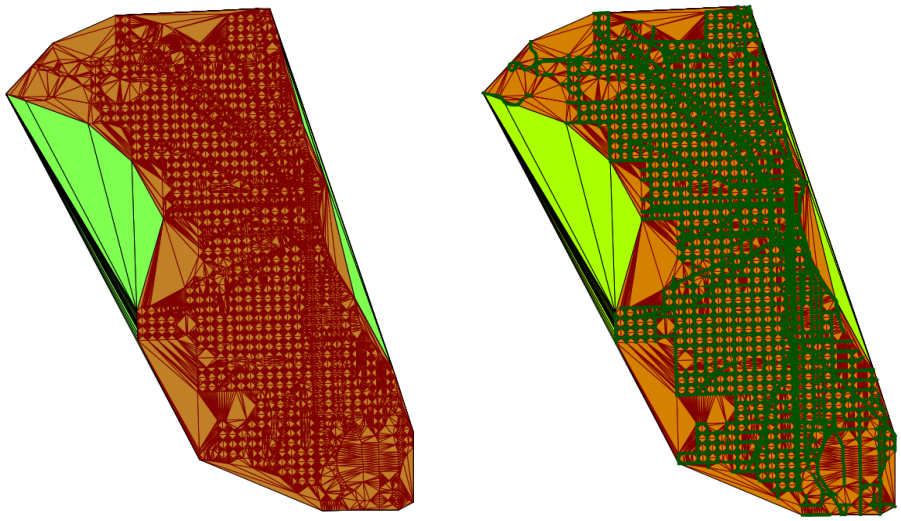


Figure 28: Triangulation and removal of big triangles – Chicago's roads

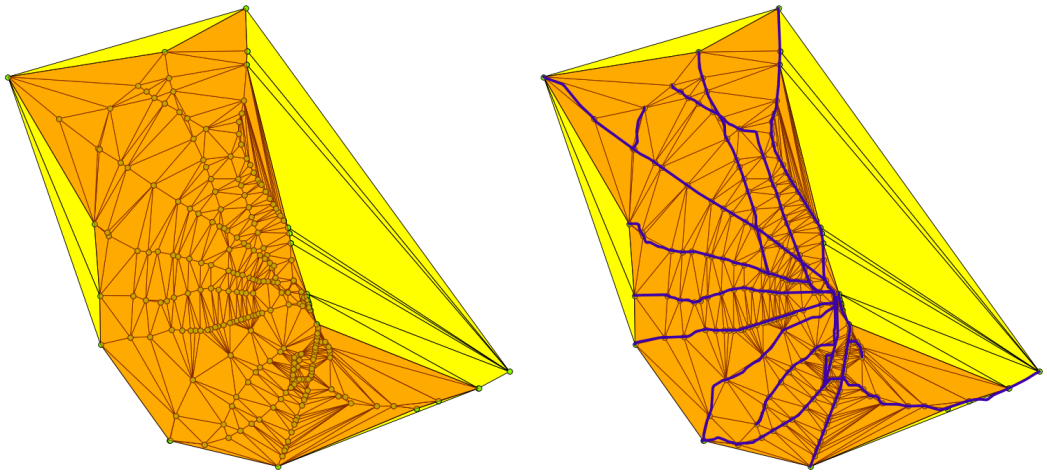


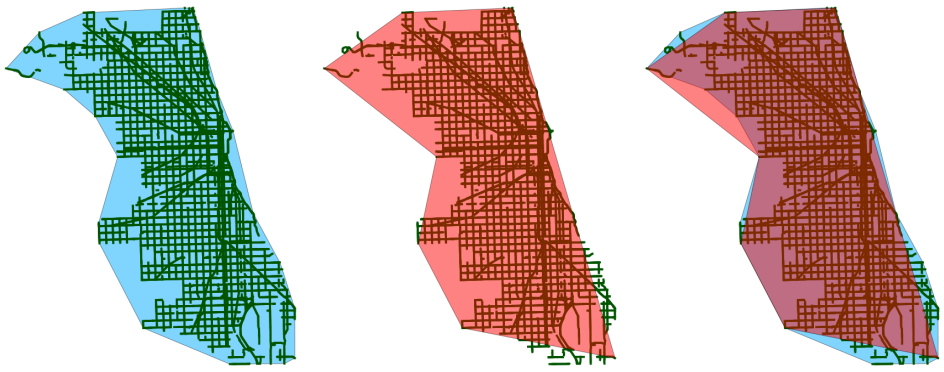
Figure 29: Triangulation and removal of big triangles – Chicago's railroad network

Table 2: Algorithm 1-a

Algorithm 1-a: (Triangle removal)

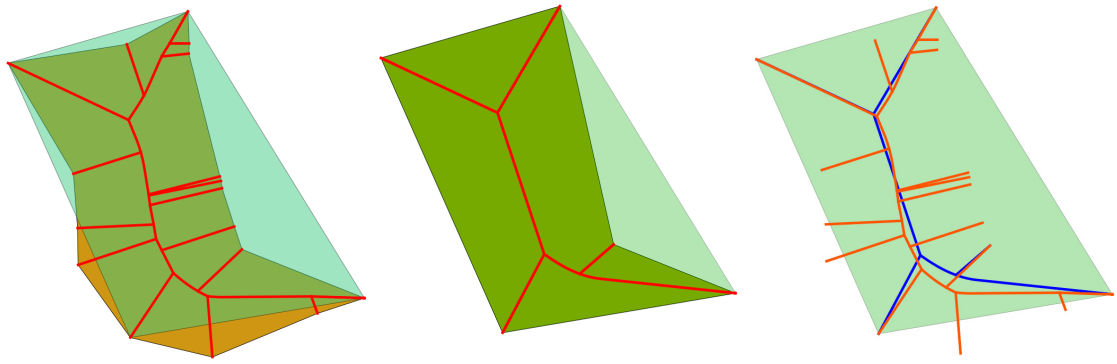
Input: Triangulation of the road network's point set
Output: Envelope of the point set
 	
find triangle with largest area;
find triangle with longest perimeter;
for each triangle of the triangles set
  if *Area*(triangle) > const. × largest area
  for each point ∈ point set
  if point ⊂ triangle
  Store(triangle);
  break;
  else if PerimeterLength(triangle) < const. × longest perimeter
  Store(triangle);
Combine(stored triangles);
Output(envelope);

The above described technique for finding the envelope, although it achieves the desired level of accuracy we want, it is not helpful when trying to find the skeleton. The reason is that, as can be seen in the above figures, the number of points forming the envelope can become large very quickly, especially if the network's shape is not simple. This causes the skeleton to have too many branches. As a result, the number of terminal and junction points of the skeleton increases significantly. This leads to more combinations to generate and more added complexity to the main problem. This complexity is actually not required as the main reason for finding the skeleton is to place the BS's. Not being power constrained as the sensors, the BS's can tolerate a slight displacement from their optimal coverage position. Also, the deployment of the sensors in a gradual manner radially outwards from the BS's helps is mitigating any BS access-related issues (from the sensors' perspective; as the BS will be powerful enough to send messages to the CC as well as the adjacent BS's). For these reasons a simplified envelope would be sufficient to specify the locations of the BS's. Thus, the previously generated



(a) (b) (c)

Figure 30: Envelope simplification: (a) Original envelope. (b) Douglas-Peucker simplified envelope. (c) Comparison



(a) (b) (c)

Figure 31: A comparison between the generated BS locations (transparent polygon corners): (a) Original envelope, skeleton, and BS locations. (b) Simplified polygon, skeleton, and BS locations. (c) Comparison (coincident BS locations).

envelope is simplified using Douglas-Peucker's algorithm as shown in Fig. 30. A tolerance value of $0.05\,L^{T}_{Max}$ was used. Again, this is a design parameter. Fig. 31 shows the resulting BS locations and the difference in the complexity of the resulting skeletons using the original and the simplified envelopes. In that figure we used Chicago's railroad network for the test. It is clear that the locations of the BSs are exactly the same for the two cases. The reason is that the removed branches are the smaller ineffective ones in the determination of the area that covers most of the network. Therefore, the simplification is worth doing.

Although, we use the simplified envelope for getting the skeleton followed by the locations of the BSs, the original non-simplified envelope is still of use. It is used to find the centroid which is used for placing the CC, as this will be more accurate. Also, when placing the sensors, it is better to use that envelope as it is a better enclosure of the network links than the simplified version. Hence, we resort back to the original envelope

when placing the sensors. Referring back to Fig. 26-b, one can see this in action. Now, that all parts of the first method have been covered, the algorithm for Method I of sensor deployment is provided below in Table 3 and the sub algorithms in Tables 4 and 5.

Table 3: Algorithm 1

Algorithm 1: (Sensor Deployment: Method I)

Input: Road network links' shapefile
Output: Deployment radii, inter-radius sensor densities, locations of the CC and BS's, and a shapefile for display purposes

1 *Extract*(points forming the links);
2 *Triangulate*(extracted points);
3 use **Algorithm 1-a** to find point set's envelope;
4 *Store*(envelope);
5 use Douglas-Peucker's algorithm to simplify the envelope;
6 *Skeleton*(simplified envelope);
7 use **Algorithm 1-b** to find BS locations' polygon; // *vertices are the locations*
8 *Centroid*(envelope); *// CC location*
9 **for** each BS
10 use **Algorithm 1-c** to find sensor densities/locations around the BS;
11 use **Algorithm 1-c** to find sensor densities /locations around the CC;
12 *Output*(CC location);
13 *Output*(BSs' locations);
14 *Output*(deployment radii and inter-radius sensor densities);
15 *Display*(sensor distribution); *// using the sensor locations' shapfile*

Table 4: Algorithm 1-b

Algorithm 1-b: (BSs' locations)

Input: Simplified Envelope's skeleton
Output: BS's polygon

```
1   find terminal and junction points of the skeleton;
2   threshold := const.;                                  // design parameter
3   combinationSize := 3;
4   do
5     for all combinations of size combinationSize        // in the point set just found
6       GenerateCombination();
7       CreatePolygon();                                  // using the generated combination
8       for each point in the terminal and junction point set
9         find distance to the created polygon;
10          Store(distance);
11        Sort(distances);
12        if maxDistance < threshold
13          Output(polygon);
14            return;
15      combinationSize += 1;
16   Forever;
```

Table 5: Algorithm 1-c

Algorithm 1-c: (Sensor locations)

Input: BS (or CC) location

Output: Sensor locations' shapefile, deployment radii, and inter-radius sensor densities

```
1   DeploymentSmoothnessFactor := const.      // design parameter (e.g. =10)
2   deploymentStartRadius := 0.5 × transmissionRange;
3   for each BS/CC
4     GenerateRadii(deploymentSmoothesFactor);     // # of radii = smoothness
5                                                  // radii values are according to Eq. 39
6   for each BS/CC
7     for each ring                                // # of rings = # of radii - 1
8       GenerateRandomSensors();                   // # of sensors is according to Eq. 40
9   Output(sensor locations to shapefile);         // for display purposes only
10  Output(deployment radii and inter-radius sensor densities);
```

4.2.2.2.2. Method 2

In the previously discussed deployment method, a very important consideration was neglected; link ranks. Not all the links have the same importance and criticality for both the civilians and HS personnel, and as a result, to the attacker. Therefore, when protecting the roads of a transportation network, it is necessary to account for this criticality and use some means to address it. Important links should receive more attention. This means that more monitoring and surveillance are required. From WSN's perspective, this means more sensors (for additional accuracy), the need for longer lives for sensors near the important links, and higher data sensing rates. This can be simply translated as more powerful nodes and a higher density near the important links. For this reason, an enhancement has been made to the first algorithm to address this issue.

The solution to the above mentioned problem is achieved through the use of electrostatic field theory. We believe that there is a great similarity between the problem under consideration, sensor deployment, and the electric fields generated by a set of charges at least in the transportation network scenario. In electrostatics, an electric charge generates an electric field in its proximity and that field decreases as we move away from the charge. As is well known, charges can be positive; having the field lines leaving the charge, or negative; with field lines directed inwards towards the charge. Coulomb's Law states that a charge will always exert force on another charge, even when the charges are separated by a large distance (Guru and Hiziroğlu, 1998). For a unit positive test charge in space, the electric field at that charge due to a single point charge, using Coulomb's Law, is given by Eq. 43:

$$\vec{E} = \frac{1}{4\pi\varepsilon_0}\frac{Q}{r^2}\hat{r} \tag{43}$$

Where:

$\vec{E}$: Electric field intensity

Q: Charge magnitude (C)

$\hat{r}$: Unit vector pointing from Q to the test point

ε_0: Permittivity of the medium (8.85×10^{-12} Fm^{-1})

r: Distance between Q and the test point

When there are n charges, the field at the test charge is the superposition of the fields due to all point charges, and is given by equation 44. Fig. 32 shows an example of a set of four positive point charges and their net electric field at the middle test point.

$$\vec{E} = \sum_{i=1}^{n}\vec{E}_i = \sum_{i=1}^{n}\frac{1}{4\pi\varepsilon_0}\frac{Q_i}{r_i^2}\hat{r}_i \tag{44}$$

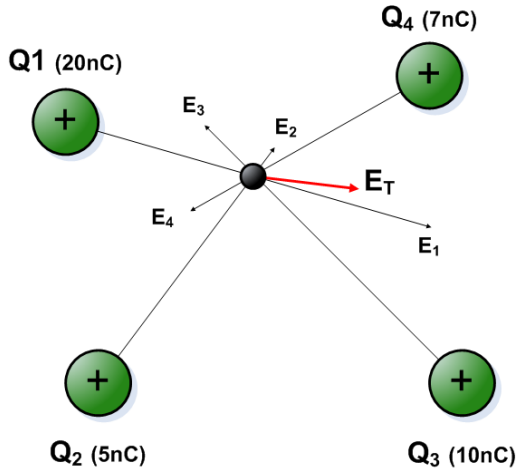
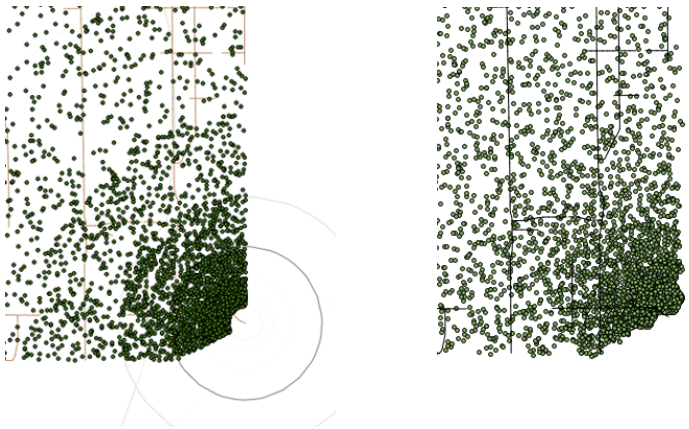


Figure 32: Effect of the electric field due to a set of point charges one a test point (the middle back point)

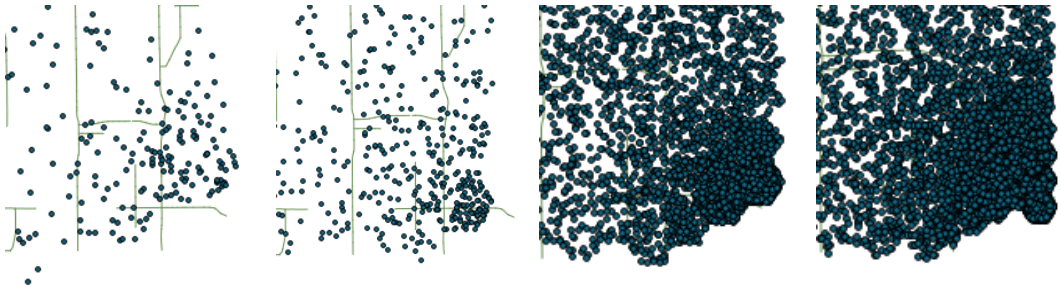
We exploit this concept to account for the important links in the design of the deployment algorithm. First, for the second algorithm to have a gradual deployment around the BS's and the CC as was the case in the other algorithm, a positive charge of a certain magnitude is assigned to each BS and to the command center. In the testing stage we used equal values of charge for all the BSs and for the CC. After that, the network area is divided into small hexagonal cells that only cover the previously found envelope (the original precise envelope in Method I, not the simplified one). These cells act as a grid covering the whole network. In the next step, Coulomb's Law is used to find the net electric field due to these charges (using Eq. 44) at the center of each grid element, i.e. the center of each hexagon. The final step includes mapping the resulting field value at each element to a number of sensors to be deployed randomly within that element. The above steps will result the desired gradual deployment. Fig. 33 shows the difference between the two techniques.



(a) (b)

Figure 33: Difference between the deployment gradient in (a) Method I and (b) Method II

Although Fig. 33-b might seem less appealing, the used technique is actually better than the one used in Fig. 33-a. The reason is that as the grid-cell size becomes smaller and smaller, the gradient becomes smoother. More precisely, having small enough (relative to the average link length in network under consideration) grid cells, makes the differences in the numbers of generated sensors moving from one cell to the other very slight. For example, in the above figure, the grid element size used is comparable to the length of the link being protected. This makes the deployment shown in Fig. 33-b unsuitable for this granularity. However, if the grid-cell size is much less than the link length, the transition from the area very close to the BS outwards becomes unnoticeable. Fig. 34 shows the gradient of the deployment for grid-cell side lengths of 3000, 2000, 1000, and 600 from left to right.



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

Figure 34: Deployment gradients for grid-cells' side length of (a) 3000, (b) 2000, (c) 1000, and (c) 600.

The algorithm then uses the same above technique to protect important links. The difference is that in the case of transportation network links, we have line segments describing them instead of just points or locations as was the case with BS's and the CC.

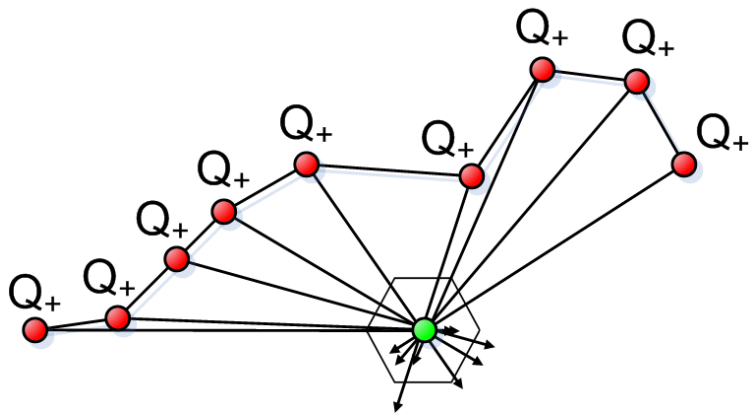


Figure 35: Charges assigned to a link's junction and end points, and the resulting effect on a grid cell's center point.

To overcome this problem without computationally complicating the problem, charges of the same magnitude are assigned to each junction point connecting any two line segments in the set of line segments that form the roads. Otherwise, we would have had to consider line charges and do integrals over the lines, which may result numeric overflows, memory run-outs, and other dramatic increases in the computational complexity. An example link along with the associated charges is shown in Fig. 35. The figure also shows the effect of these charges on a test point; a grid cell's center. After assigning charges to transportation links, the problem is treated similarly to what was done before; the field value is found at the center of every grid cell and sensors are placed accordingly. An illustration of the resulting deployment is shown in figure 36. The only remaining question is how to consider the ranks? The answer to this question is to give each link in the highly ranked links, chosen to be ten in this work, a charge value proportional to that rank. By doing this, the rank-1 link will have the highest charge value and consequently the highest associated field. Rank-2 link will have the second highest field, and so on.

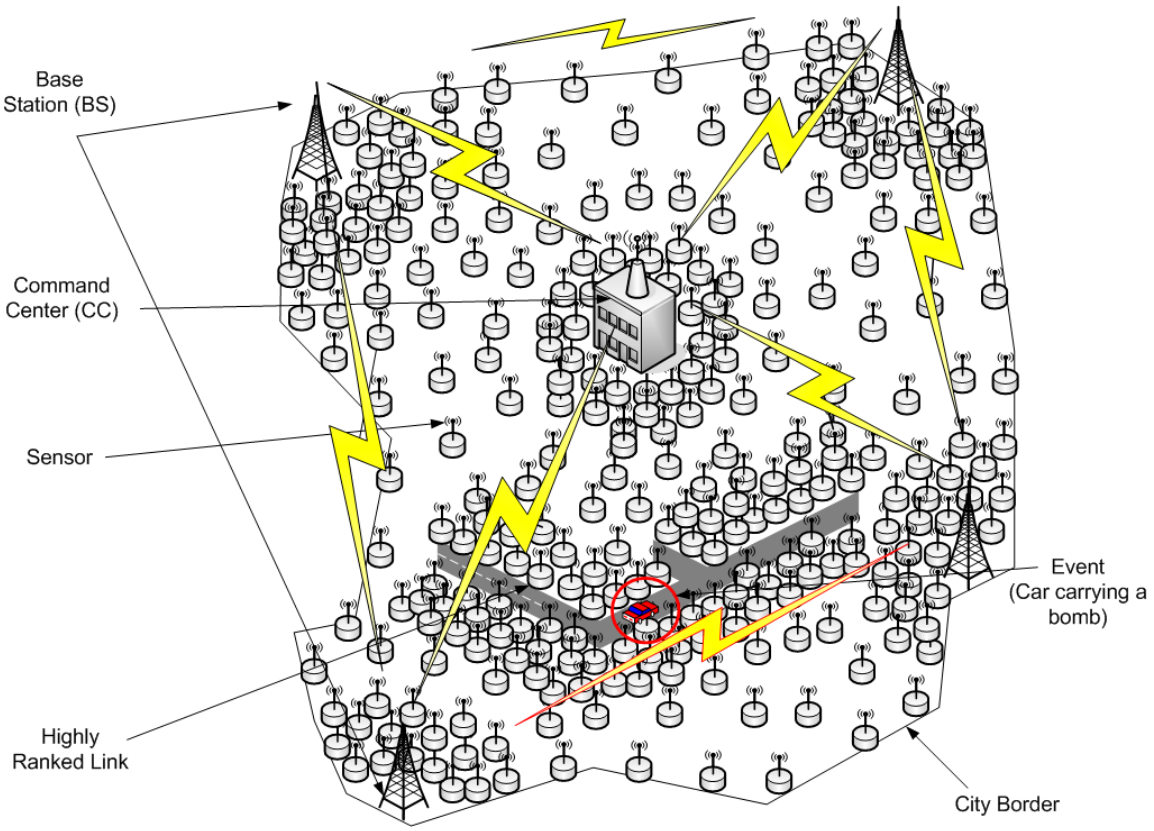


Figure 36: Illustration of Method II of sensor deployment

This will cause the deployment gradients to depend on the rank, and therefore, the highly ranked links will have higher concentrations of sensors near them, while the other low rank ones or the ones that have not been ranked will get lower densities of sensors. For the links that have not been ranked, the sensor density will be governed by the charges of the BS's, the CC, and other close links that has been ranked. Fig. 37 shows an example of this deployment technique as applied to Chicago's road network, and Fig. 38 shows a sample deployment info output.

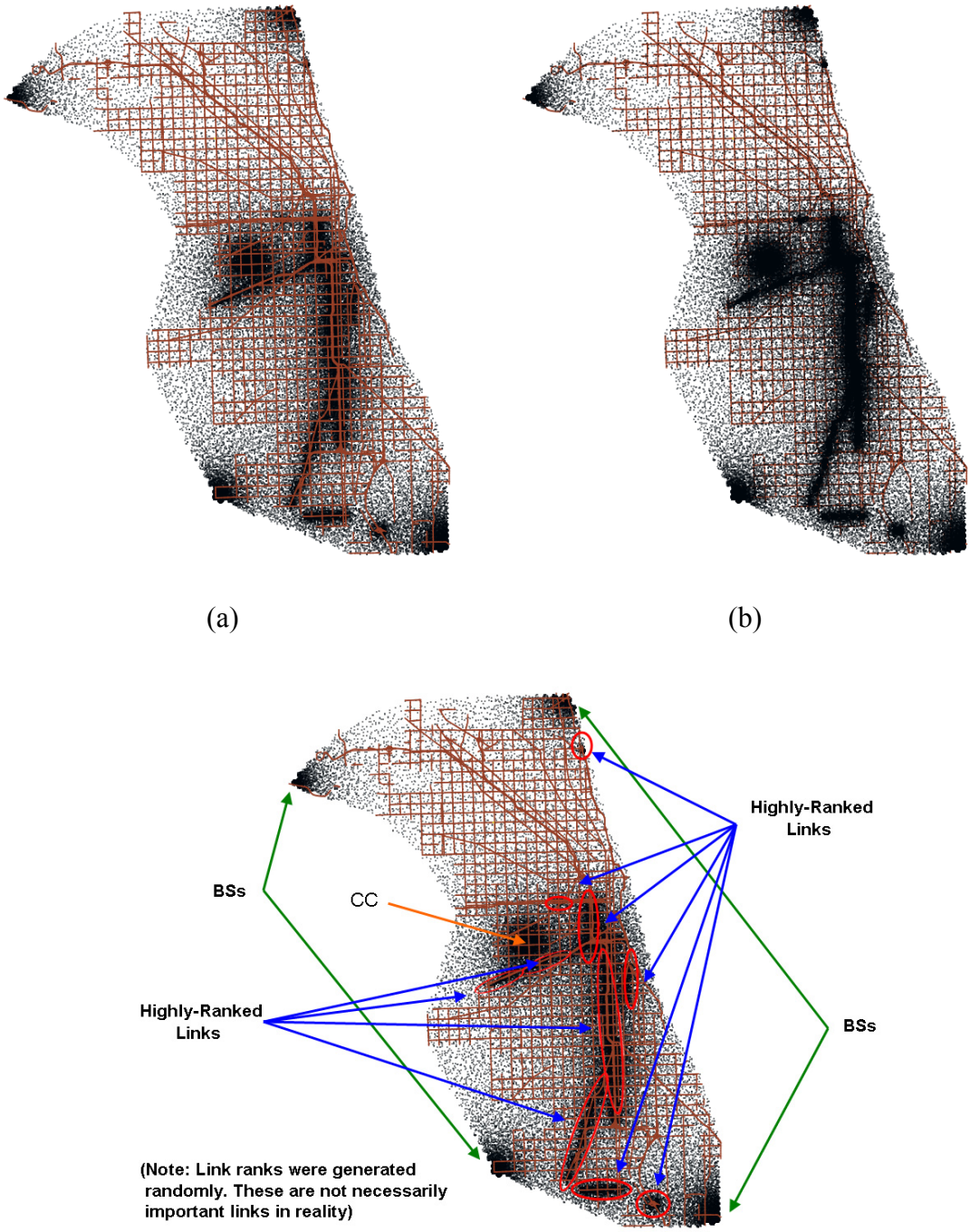


Figure 37: Deployed sensors on Chicago's road network using deployment method II: (a) Links on top; (b) Sensors on top; (c) Elements highlighted.

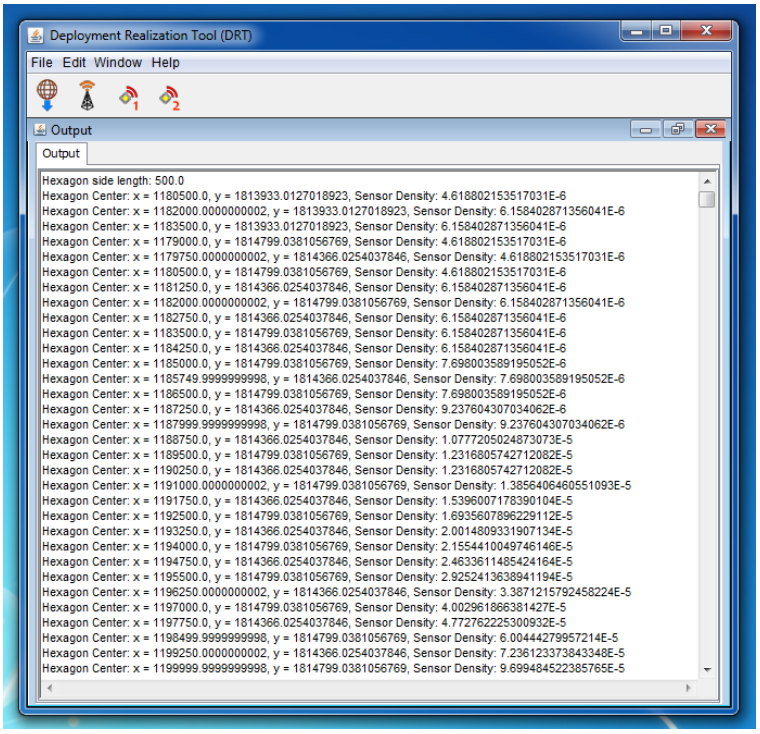


Figure 38: Deployment method II: sample output

Now, that the basics needed for understanding how the second method works were covered, we move forward to the details of the algorithm. The second deployment method has the same main preparation steps as the previous one, i.e. steps 1-8 in algorithm 1. The only difference is in the step associated with the placement of sensors. Therefore, we can summarize the second method as follows: 1) first, the envelopes, both the precise and the simplified, of the network are found ‘steps 1-5 in algorithm 1’, 2) the skeleton of the simplified envelope is then found ‘step 6’, 3) BSs’ and CC’s locations are found using algorithm 1-b ‘steps 7-8’, 4) a hexagons grid covering the precise envelope is then generated using a pre-specified hexagon side length, 5) charges are assigned to BS and CC locations, 6) charge values, to be assigned to the links proportionally to their ranks, are calculated based on the number of ranks considered. 7) the charges are then

assigned to the link points, 8) the field due to the set of all point charges; BS's, CC, and links' point charges, is found at each grid cell, 8) the field value at each grid cell is then mapped to a number of sensors that are randomly deployed in this cell with a uniform distribution. Table 6 lists Algorithm 2.

Table 6: Algorithm 2

Algorithm 2: (Sensor Deployment: Method II)

Input: Road network links' shape file

Output: Sensor deployment locations shape file, grid cells' center locations, and sensor densities

```
1   hexagonSideLength := const.      // Design parameter based on the average length of
2                                    // a network link
3   ccCharge := const.          // Design parameter; the charge assigned to the CC
4   bsCharge := const.          // Design parameter; the charge assigned to each of the BSs
5   Extract(points forming the links);
6   Triangulate(extracted points);
7   use Algorithm 1-a to find point set's envelope;
8   Store(envelope);
9   use Douglas-Peucker's algorithm to simplify the envelope;
10  Skeleton(simplified envelope);
11  use Algorithm 1-b to find BS locations' polygon;
12  Centroid(envelope);     // CC location
13  CreateGrid(envelope, hexSideLength);     // create a hexagonal grid covering
14                                           // the envelope
15  AssignBSCharges(bsCharge) ;
16  AssignCCCharge(ccCharge) ;
17  for each rank                            // the first 10 most important links
18    CalculateChargeValue();
19    Store(charge value);
```

```
20  for each link
21    for each point on the link
22      assign the corresponding charge value based on link's rank;
23  for each grid cell
24    for each point charge that is stored      // stored point charges include the CC's point
25                                              // charge, BSs' point charges, and the point
26                                              // charges of all highly ranked links
27      compute the electric field at the center of the grid cell;
28      map the field value at that cell to a number of sensors;    // the integer floor of the
29                                                                  //  value of the field was used
30      generate that locations randomly;       // uniformly distributed
31      Store(sensor locations);
32  Output(CC location);
33  Output(BSs' polygon);     // corners are the BS locations
34  Output(grid cells' center locations and sensor densities);
```

4.3. Simulation Environment and Implementation

In this section, the implementations of the attacker-defender model both in VenSim and in Java are discussed. Followed by that is the Java implementation of the sensor deployment algorithms and an associated tool designed to serve as an assistive tool to HS personnel in getting the information needed for the actual deployment.

4.3.1. Simulating Attacker-Defender Interaction

As discussed previously, system dynamics is very well suited for modeling complex systems. There are several system dynamics' model development and analysis software environments available such as DYNAMO, IThink/Stella, PowerSim, VenSim, and others[1]. Ventana Systems Inc. provides a free personal learning edition of VenSim called

[1] http://www.vensim.com/sdmail/sdsoft.html

VenSim PLE. This and other commercial versions provide a very good GUI that facilitates the design and analysis process. VinSim PLE was found sufficient for the purpose of this work. The attacker-defender model was first designed and analyzed in VinSim, and then it was implemented in Java to have full integrity with the other tools being developed. Actually, VenSim provides interfaces for integration with other software, but we preferred the above approach to guarantee full control. Next, we provide the details associated with the implementation of the model in both environments.

4.3.1.1. VenSim Simulation

In VenSim, models can be easily constructed graphically through the insertion of different model variables on a desktop like pane. After the variables are inserted, connections can be made, and the equations associated with each variable can be input as functions of other variables, from a spread sheet file, or by other means. One good thing about VenSim is that the model will not work until all variables' units are consistent. We believe this a one way of validating a model as the designer will be checking model correctness during the design process through two steps; the correct modeling of each variable and the verification of this correctness by the proper units connecting this variable to the models of other related variables. VenSim also provides some useful analysis tools that ease the analysis of the designed model as will be seen in the discussion below.

The first sub-model, the HS perception model, was implemented in VenSim as shown in Fig. 39. Each of the variables shown in figure has an underlying equation as described in section 4.2.1.2. VenSim allows the designer to enter the equation through the equation editor by double clicking the variable in the equation mode. For example, the

equation for the HS-S probability of attack success appearing in the equation editor is shown in Fig. 40.

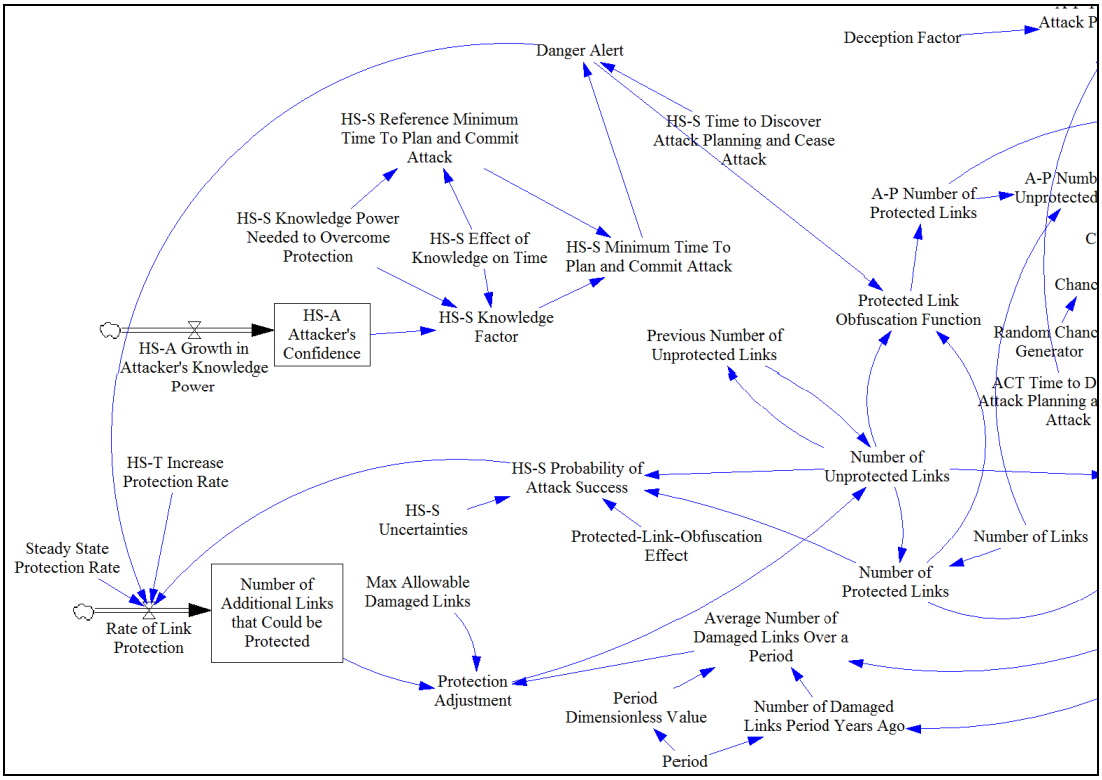


Figure 39: HS Perception Model Implementation in VenSim

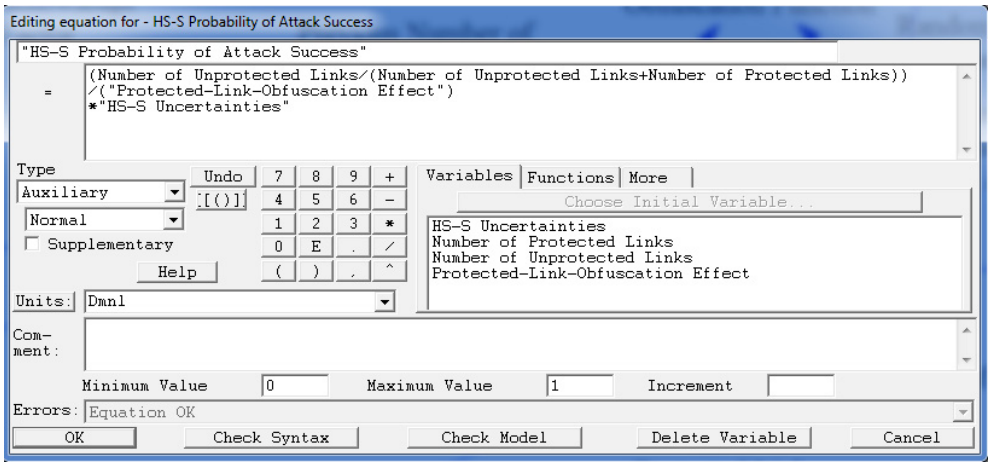


Figure 40: Specifying the equation for the HS-S probability of attack success

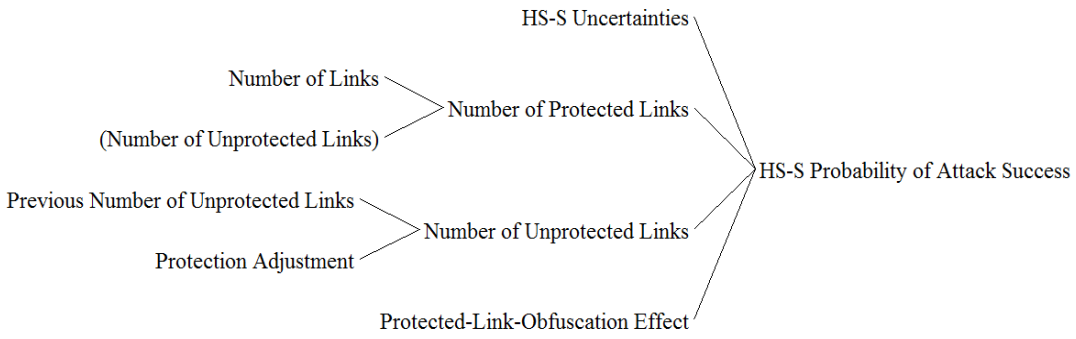


Figure 41: Causes Tree of the HS-S probability of attack success

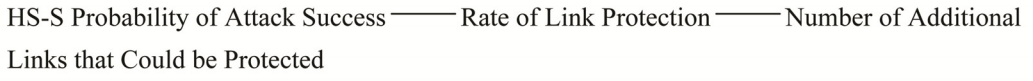


Figure 42: Uses Tree of the HS-S probability of attack success

Using the so called ‘causes tree’ enables us to show the model variables that ‘caused’ the variable under consideration or, in other words, that led to the change in the value of that variable. As the probability of attack success from the HS agency’s point of view is the main output in the first sub-model, we study it more closely. Fig. 41 shows the causes tree for that variable. As can be seen, it is a direct function of the HS-S uncertainties, the number of protected and unprotected links, and the protected link obfuscation function. It is also an indirect function of the variables on the second level of the tree (leftmost group of variables).

In addition, a ‘uses tree’ shows the uses of the variable under consideration by other variables. The uses tree of the HS-S probability of attack success is shown in Fig. 42. The tree shows that the rate of link protection is a function of the HS-S probability of attack

success. Similarly, the number of additional links that could be protected is a function of the rate of link protection.

The HS perception model is connected to the other two sub-models through the variables discussed in section 4.2.1.2. This can be seen through the trimmed right side of Fig. 39. The attacker perception model and the real world model are shown in Figs. 43 and 44 respectively. It should be clear how the three sub-models interact as variables included in each sub-model affect, in either a direct or an indirect way, the variables in the other two sub-models. As in the HS perception model, the trimmings on the sides of Figs. 43 and 44 are due to the connectivity to the other sub-models.

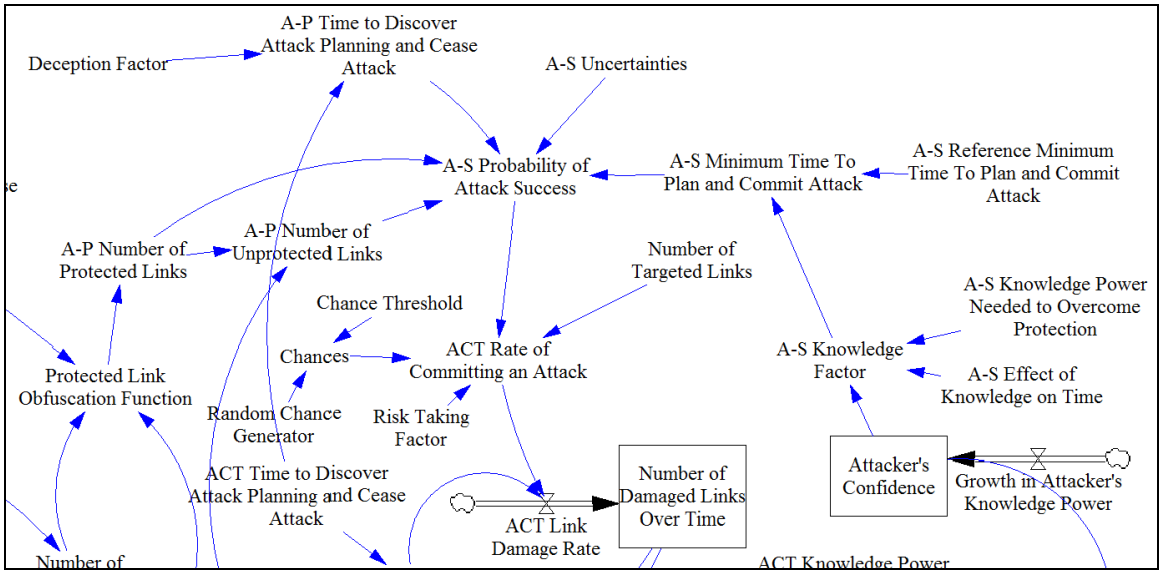


Figure 43: Attacker Perception Model implementation in VenSim

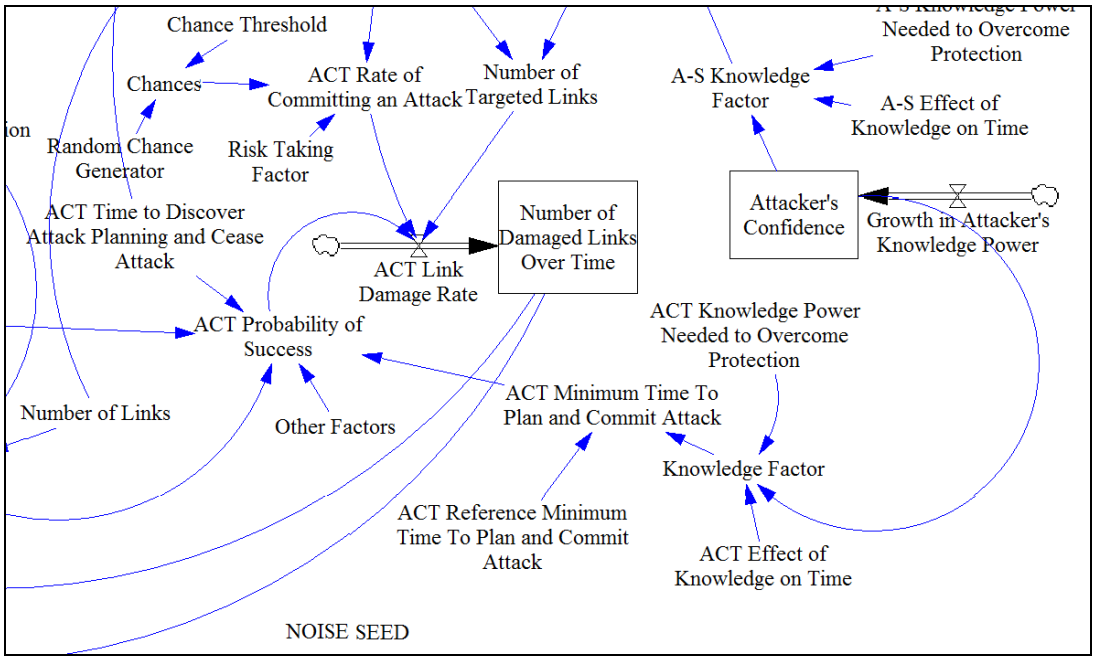


Figure 44: RW Model implementation in VenSim

4.3.1.2. Java Implementation

In addition to the implementation in VenSim, the attacker-defender interaction model was implemented in Java. The purpose of re-implementing it in Java is to have greater flexibility and to support the integration of all developed tools. The implementation followed the system dynamics' sequence of operations. This sequence is illustrated in the flow chart given in Fig. 45. Java, being an object oriented language, is very suitable for the purpose of that model. Each entity involved in the interaction was represented by a class, and instances were drawn. For example, the homeland security agency, attacker, real-world, and interaction classes were created. This is really helpful as it allows for the extensibility of the model. If, in the future, it is needed to model different attacker types, the collaboration of different HS agencies, or parallel interactions, it would be easy to instantiate different objects of these classes as necessary. The UML diagram of the model

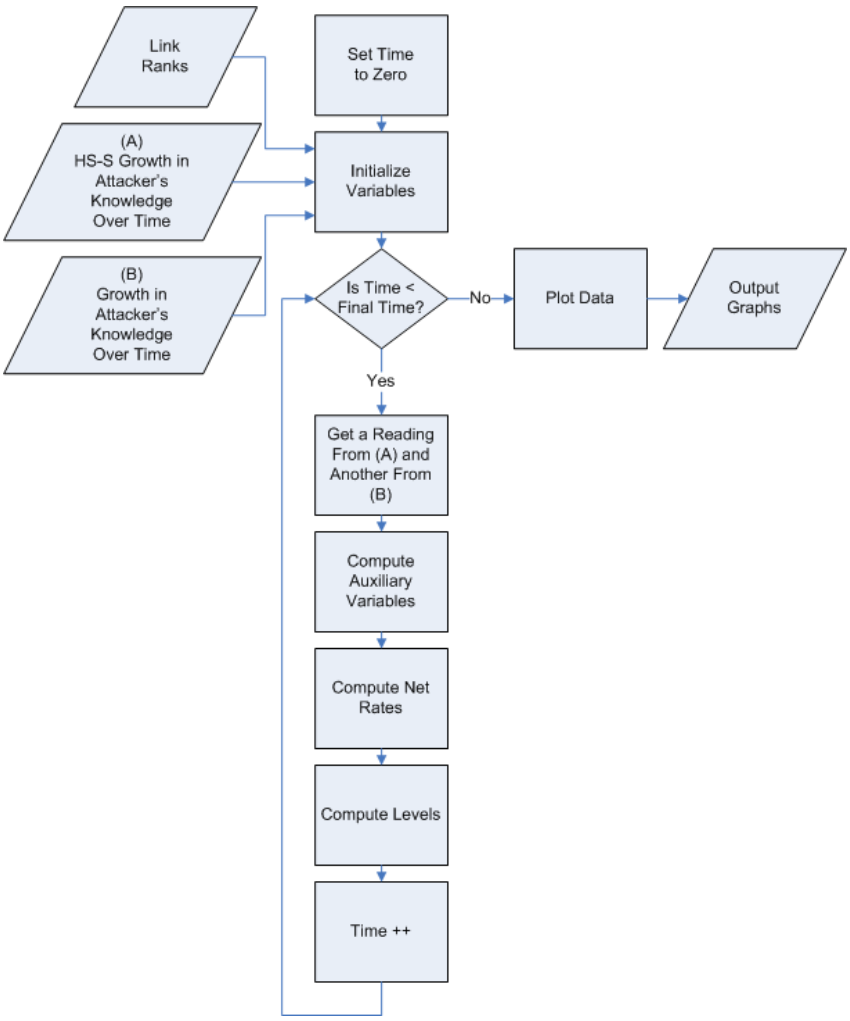


Figure 45: A mapping of System Dynamics' sequence of operations to the Java implementation

is shown in Fig. 46. Please note that not all the variables and methods are shown due to space constraints.

In this implementation, the HS agency, the attacker, and the real world have their associated properties representing their respective sub-model variables. The interaction has its own properties which represent the interaction variables. The program works as follows: An interaction tester class creates instances of the interaction, HS agency,

attacker, and real world classes. The tester then associates each of the three participating entities to one another, and associates them to the interaction. It then runs the interaction and the results are output. A sample output is shown in Fig. 47.

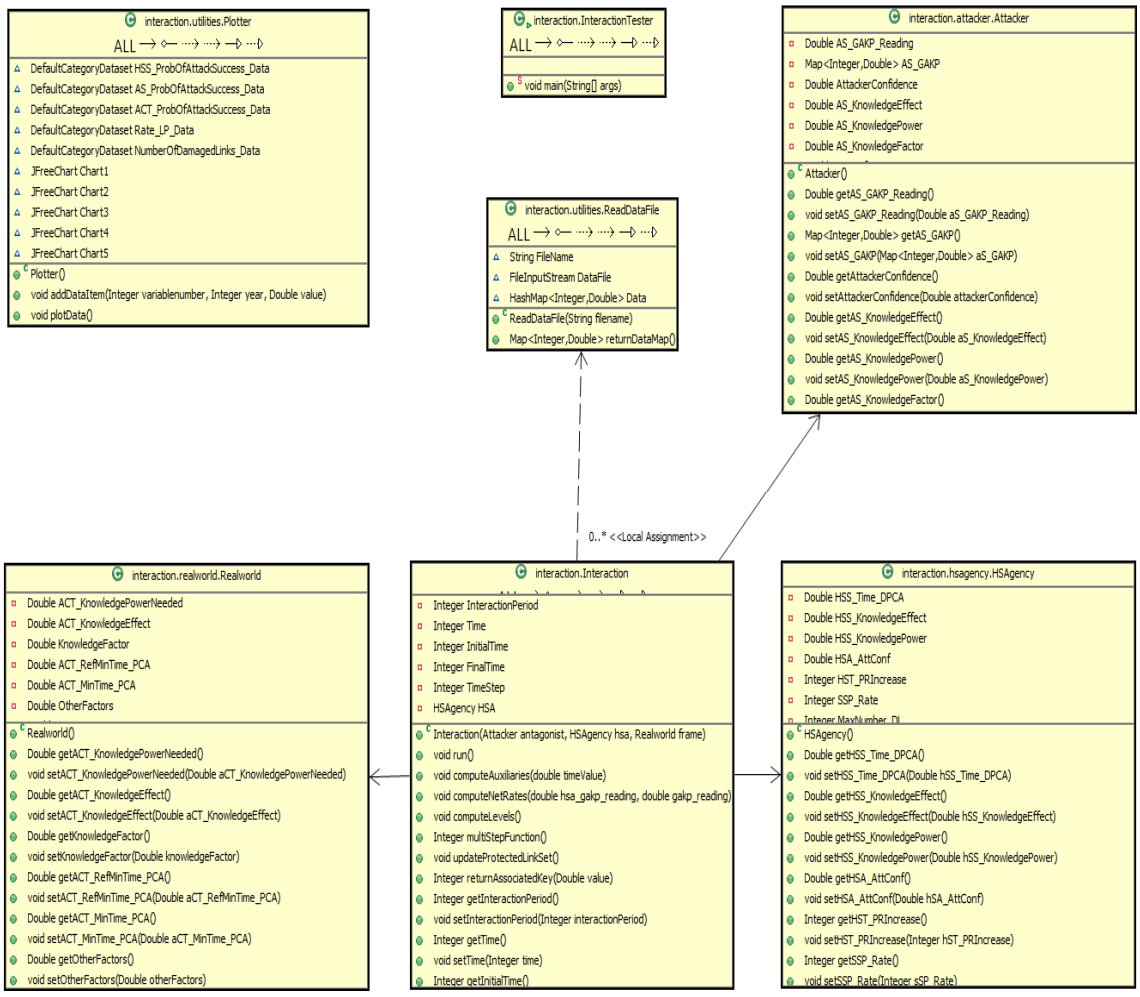


Figure 46: HS-attacker model UML diagram

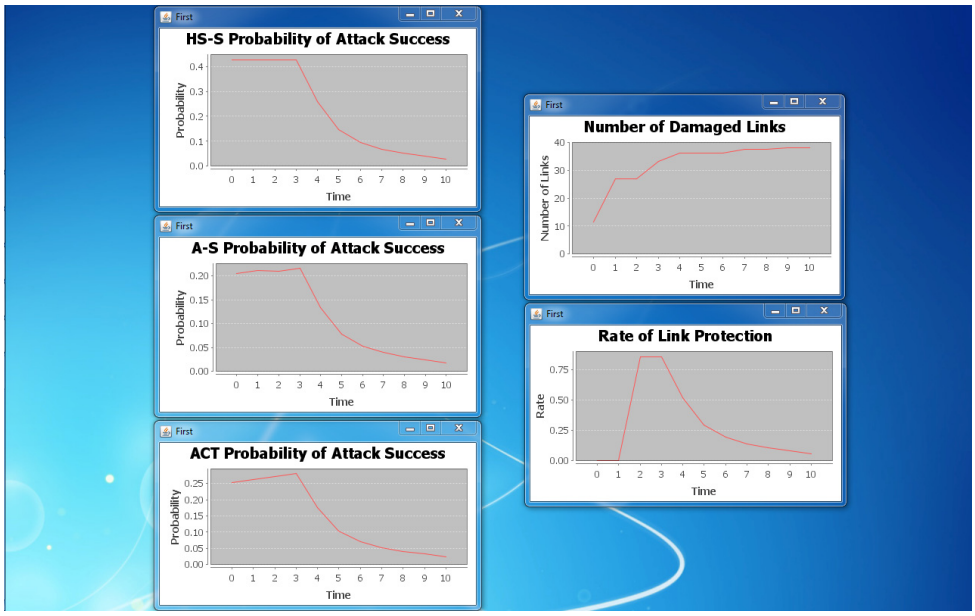


Figure 47: Sample output of the Java implementation of the HS-attacker interaction model

4.3.2. Sensor Deployment Tool – Java Implementation

A tool, with a graphical user interface (GUI), for deploying sensors using the two proposed algorithms was developed. Its aim is to assist HS personnel in generating deployment locations and densities for the considered cities easily. The tool makes use of GIS data to generate the deployment according to some coordinate system and overlays it on the map of the city under consideration. We call this tool the Deployment Realization Tool (DRT).

The tool was implemented in Java and made use of existing open sources libraries such as GeoTools[1] (GeoTools The Open Source Java GIS Toolkit) – for GIS date

[1] GeoTools The Open Source Java GIS Toolkit - http://geotools.org/

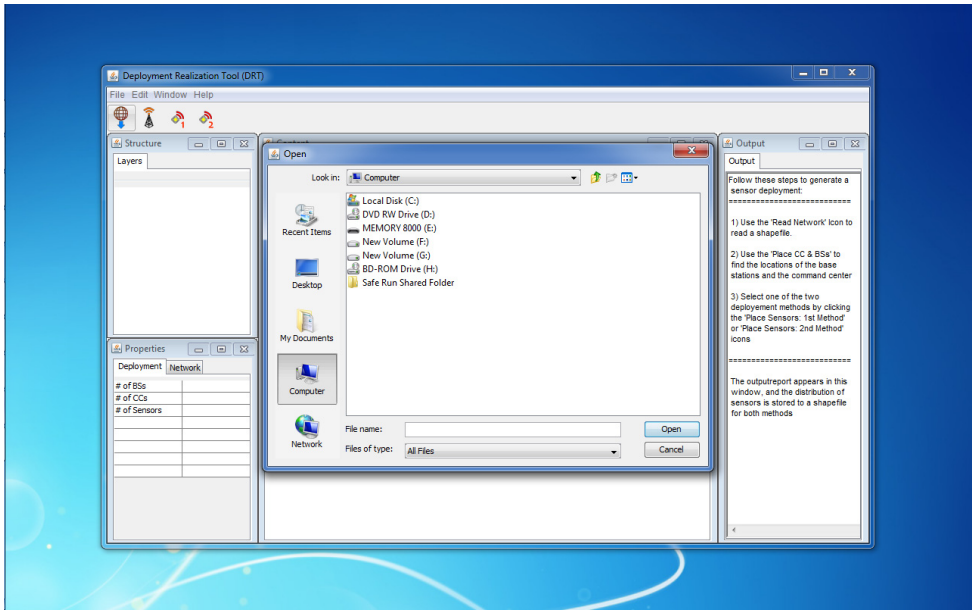


Figure 48: GUI of the Deployment Realization Tool (DRT)

handling and processing, JTS[1] (Java Topology Suite) – for some computational geometry functions, and the Skeletonizer plug-in[2] – to obtain the skeleton of the road network's envelope. Several other classes were created manually. The tool is still under development and is subject to further enhancements. In this section, we present this tool, an overview its current capabilities, and brief the additions that are planned for future work.

The GUI of the tool is shown in Fig. 48. It has a desktop layout that contains four frames; 'Structure', 'Properties', 'Content', and 'Output'. The structure frame is used currently only for showing the layers used in the project, this is done through the 'Layers' tab. In the future, it will have other tabs for showing the file hierarchy in the project and for navigating through system files to open layers. The properties window has two tabs.

[1] Java Topology Suite - http://www.vividsolutions.com/Jts/JTSHome.htm
[2] Plugins for OpenJUMP - http://sourceforge.net/apps/mediawiki/jump-pilot/index.php?title=Plugins_for_OpenJUMP

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

Figure 49: DRT's Toolbar Icons (a) Read Network (b) Place CC & BSs (c) Place Sensors: M1 (d) Place Sensors: M2

The first is called 'Deployment' and the other is 'Network'. In the deployment tab, a table lists the generated numbers of the CC, BS's, and sensors. The network tab shows the number of links in the network. These tabs will be extended to have more deployment and network parameters. The content frame shows the network being processed. It can have more than one network simultaneously in different tabs. Finally, the output frame is used to give some hints on how to use the tool when it is launched and to list the output deployment's information such as: the number of deployment radii, sensor densities, locations of grid cell centers, and the number of command centers and base stations. There are four icons in the toolbar: 'Read Network', 'Place Command Center & Base Stations', 'Place Sensors: First Method', and 'Place Sensors: Second Method'. These icons are shown in Fig. 49.

The user first reads the road network for which it is desired to generate the deployment. The input is a shape file containing the network links (roads). This is done by clicking on the 'Read Network' icon as shown in Fig. 48. The GUI then displays the network and adds it as a layer in the 'Layers' tab of the 'Structure' window on the top left. This is given in Fig. 50.

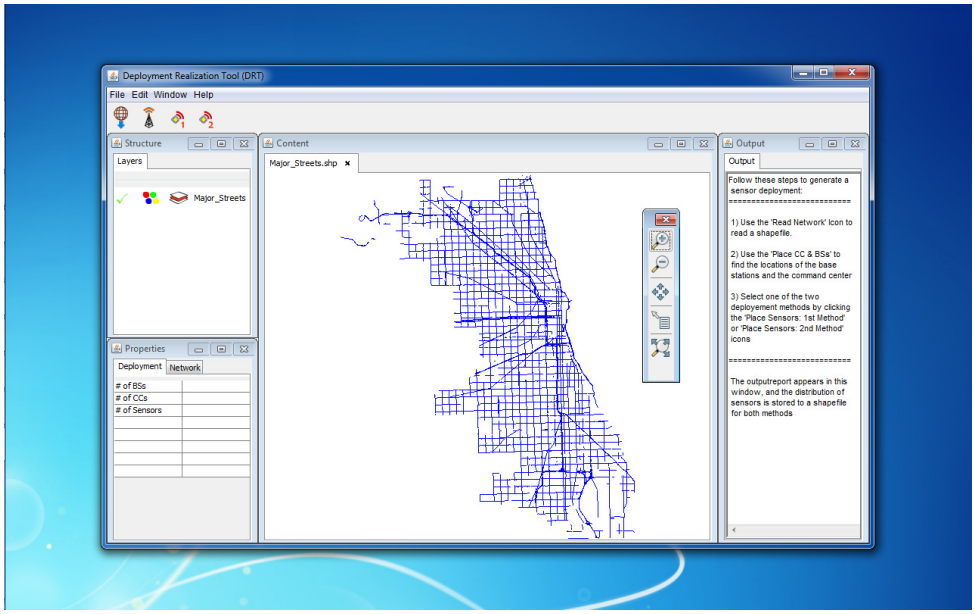


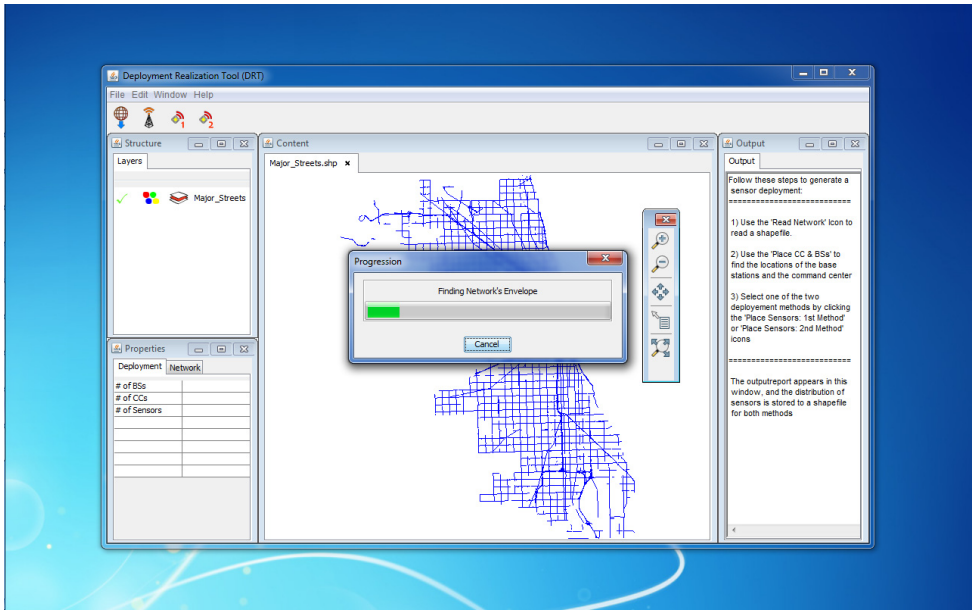
Figure 50: The read network and the associated layer in the ‘Layers’ Tab.

The next step is to deploy the Command Center and the Base Stations. This is done by clicking the “Place CC and BS’s” icon. The tool then starts executing the common part of the two algorithms. At the end of this stage, the tool will add the network’s simplified envelope, the centroid (CC location), the skeleton, and the BSs’ polygon as new layers in the ‘Layers’ tab and display them on the network as shown in Fig. 51.

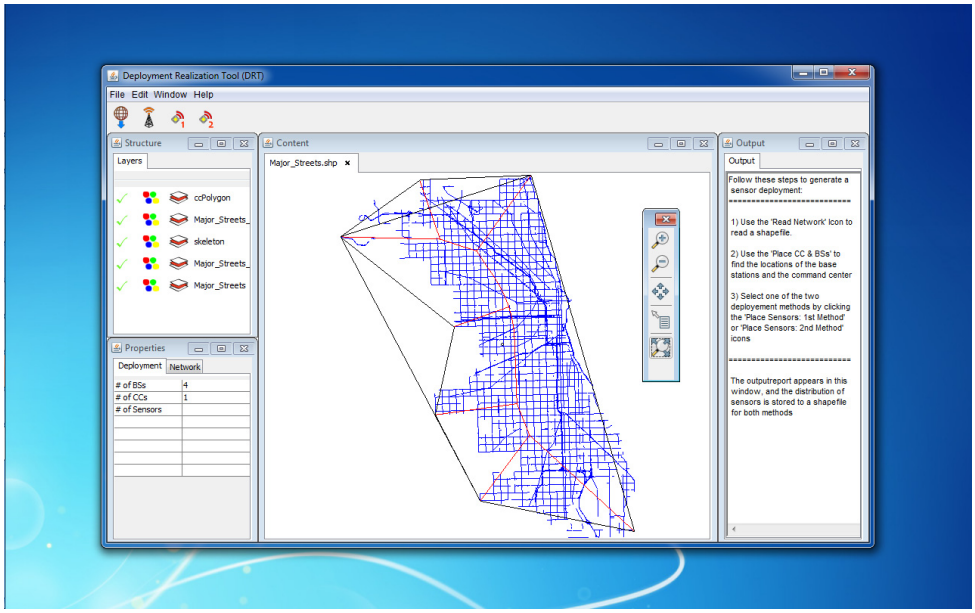
The user can next choose one of two options: sensor deployment Method 1 or 2. Suppose the user selected Method 1, the tool then executes Algorithm 1 and displays the output as shown in Fig. 52. The outputs for this and all other steps are stored in output shapefiles. This is done for each layer. The output window also displays the deployment details, including the number of BS’s, CC’s, the deployment radii, and sensor densities.

When the user chooses the second deployment method, Algorithm II is executed and the output is displayed as shown in Fig. 53. In that case, the output window also displays

the locations of grid cell centers, and sensor densities in additions to the number of CC's and BS's.

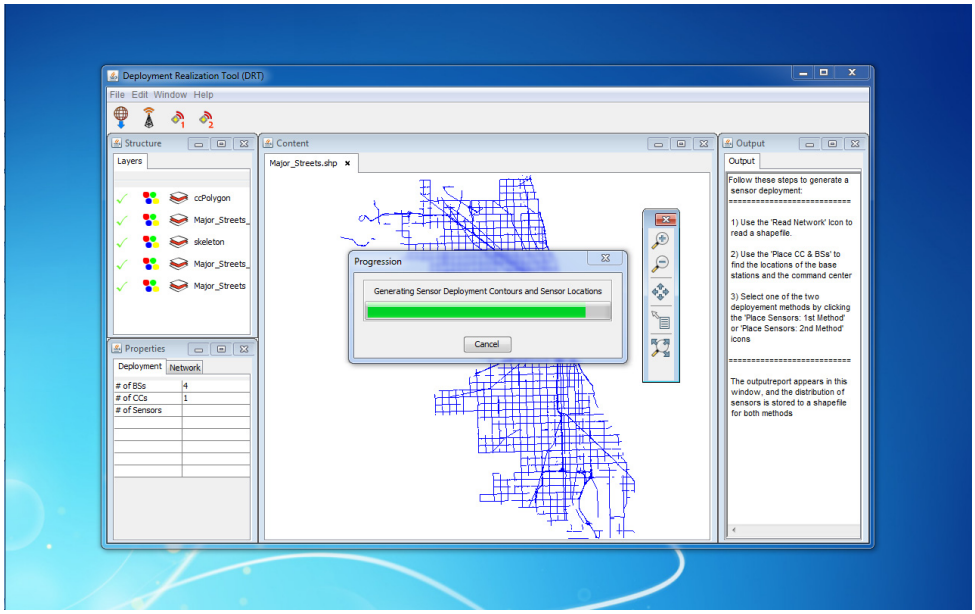


(a)

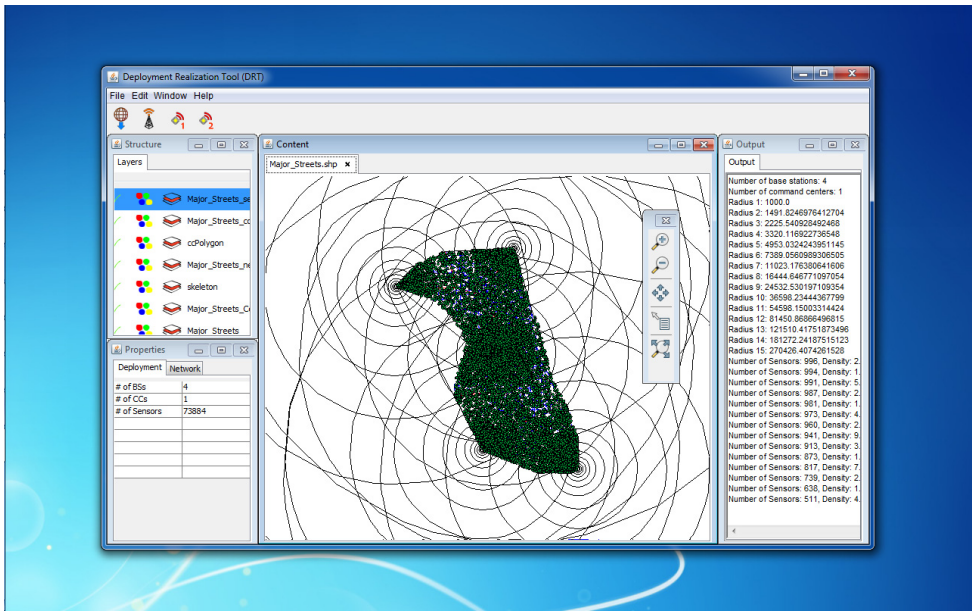


(b)

Figure 51: (a) Placing the CC and the BSs. (b) The result showing the skeleton, the simplified envelope, the centroid, and the BS polygon

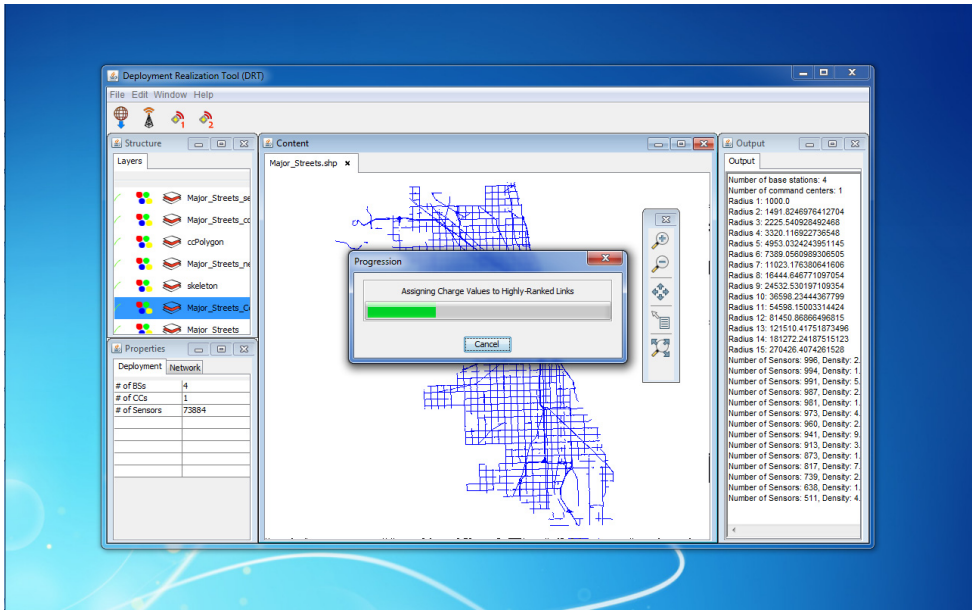


(a)

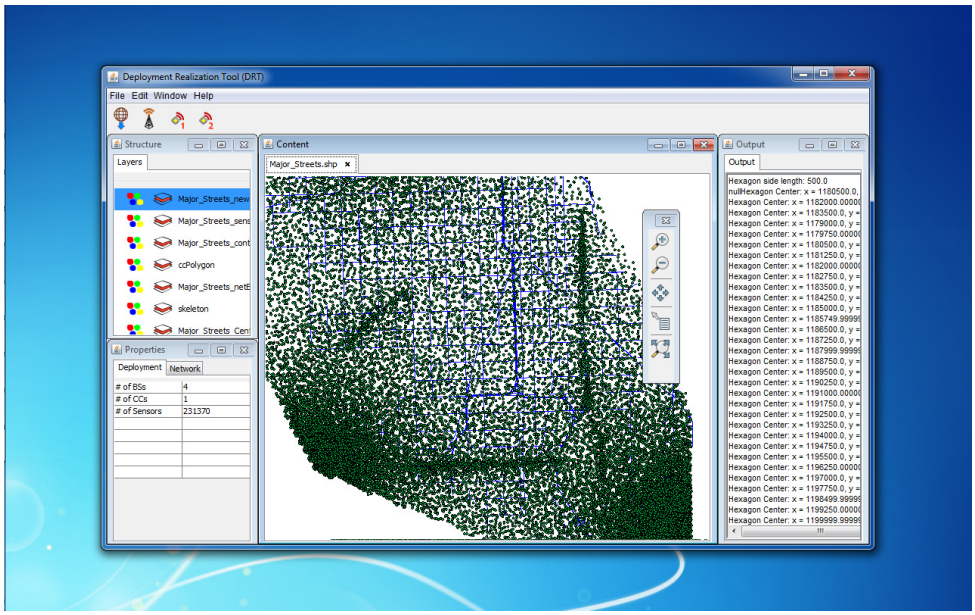


(b)

Figure 52: (a) Placing sensors using Method I. (b) The result showing the deployed sensors along with the deployment radii.



(a)



(b)

Figure 53: (a) Placing sensors using Method II. (a) The result, showing the deployed sensors. Notice that the highly ranked links receive higher sensor densities in the case.

CHAPTER V

RESULTS AND ANALYSIS

To test, analyze, and demonstrate the abilities of the proposed models and algorithms, this chapter provides two test cases for both the attacker-defender interaction model and the deployment algorithms. Sensitivity analyses are conducted and conclusions are drawn for each case. The attacker-defender tests are done first followed by the deployment algorithms tests.

5.1. Attacker-Defender Interaction

5.1.1. Test Case 1: Varying Defender's Uncertainty

In this first test case, we study the effect of lack of information on the HS agency's side on the three probabilities of attack success, on the rate of link protection, and on the number of unprotected links. For this purpose we fix all other model variables and change only the variable "HS-S Uncertainties". This variable takes the values from 0.005 to 1. The reason for the selection of 0.005 as the lower bound on uncertainty is that if it is made equal to zero the probability of attack success will be zero, which is unrealistic. Also, making it equal to zero means that the HS agency has complete information, which is a very rare case. On the other hand the upper bound is set to one to indicate that if there is a very high uncertainty about some aspects, the probability of attack success will be governed completely, form the HS point of view, by what the other contributors to that probability impose.

Case Fixtures	Units
$T_{DPCA}^{HS-S} = 0.7$	Years
$E_K^{HS-S} = 1.5 \times 10^{-0.007}$	Years / BitsOfInfo
$P_K^{HS-S} = 4$	M BitsOfInfo
$R_{SSP} = 0$	Links / Year
$R_{IP}^{HS-T} = 2$	Links / Year
$U^{HS-S} \rightarrow (we - vary - this)$	Dimensionless
$N_{DL}^{*} = 0$	Links
$O_e = 1$	Dimensionless
$p = 3$	Years
$N_L = 25$	Links
$N_{UP}^{i} = 10$	Links
$N_P = 15$	Links
$F_D = 1.654$	Dimensionless
$U^{A-S} = 0.6$	Dimensionless
$K_C = 50$	Dimensionless
$F_R = 3$	Dimensionless
$\hat{T}_{PCAr}^{A-S} = 0.25$	Year
$P_K^{A-S} = 900000$	BitsOfInfo
$P_K^{ACT} = 1$	M BitsOfInfo
$E_K^{ACT} = 4.25 \times 10^{-0.007}$	Years / BitsOfInfo
$\hat{T}_{PCAr}^{ACT} = 0.5$	Year
$F_O = 0.8$	Dimensionless
$T_{DPCA}^{ACT} = 0.5$	Year

Table 7: Values used for the variables in test case 1

Uncertainty values used in case study 1						
Value	0.005	0.204	0.403	0.602	0.801	1

Table 8: Uncertainty values used for sensitivity analysis

Table 7 lists the values used for the model parameters in this case, and the variable we change the value for. The values used for the U^{HS-S} are shown in table 8. We use an increment of 0.199 starting at 0.005 and ending at 1.

When the uncertainties of the HS agency increase from 0.005 to 1, it is noticed that the probability of attack success from its perspective (Fig. 54) increases in value, and the time range this higher probability is spanning decreases. After some year specified by the uncertainties value, the probability decreases with a faster rate as the uncertainties

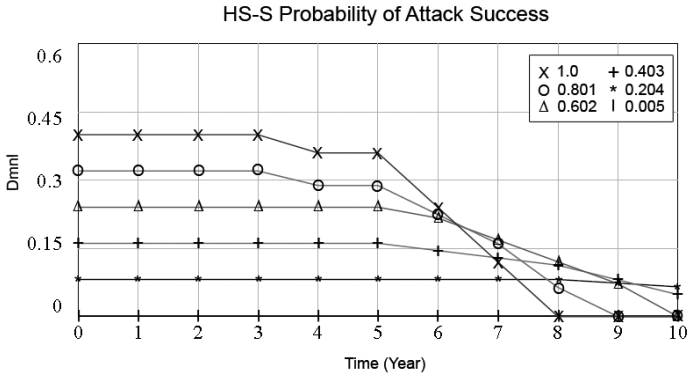


Figure 54: HS-S probability of attack success

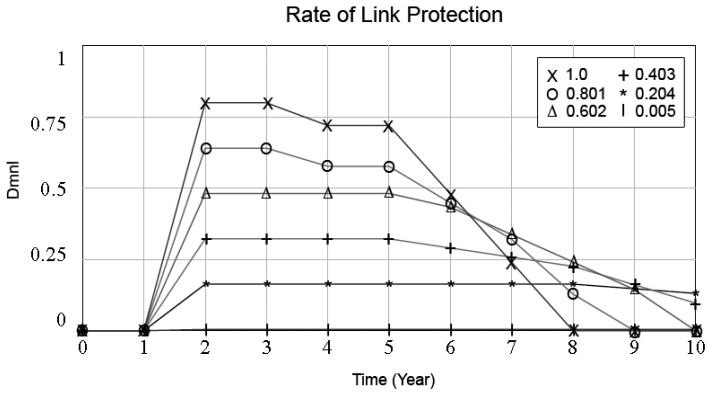


Figure 55: Rate of link protection

increase. On the other hand, decreasing the HS uncertainties causes the probability to decrease in value and to be spread over the ten year period with approximately the same probability per year. The danger alert is fired after one year from the start time as the time to plan and commit attack becomes smaller than the time needed to discover attack planning and cease attack. This happens when the knowledge needed to overcome protection is exceeded. The firing of this alert causes a change in the rate of link protection (Fig. 55) to be triggered. As can be seen from table 7, the HS tolerable increase in link protection rate is two links per year. For this value, and depending on the

current value of the probability of attack success, the increase in the rate of link protection is determined. The higher the uncertainties are, the higher the increase in link protection rate, and the higher the increase in link protection rate is, the faster that the unprotected links need to be protected by the HS agency to overcome this lack of certainty and high probability of attack success. This indicates the importance of intelligence in the protection-attack evolution process. An appropriate level of intelligence information collection should be kept along with the deployment of technology to reduce the required budget and to keep the network safer at the same time. The required level of intelligence can be inferred from the required level of certainty that needs to be kept.

It is also important to notice that the faster the rate of link protection becomes, the closer in time that rate decreases, rapidly as well, because of the resulting effect from protecting more links; which is more safety and less ACT probability of attack success (because of the increased number of protected links), and less A-S probability of attack success (due to the perception of the attacker that more links are getting protected).

Taking a closer look at the number of unprotected links over time is important as this provides the HS agency with the roadmap that it can follow to deploy the technology. Starting with the average number of damaged links over a period of time (taken here to be 3 years), it is noticed that sometimes, even when it exceeds the maximum allowable number of damaged links by the HS agency, there could be no protection adjustment i.e. no additional links would be protected. The reason for this is that protection adjustment is an indirect function of the probability of attack success from the HS agency's point of view as well. So, when this probability is reasonably low the adjustment is a fraction of a

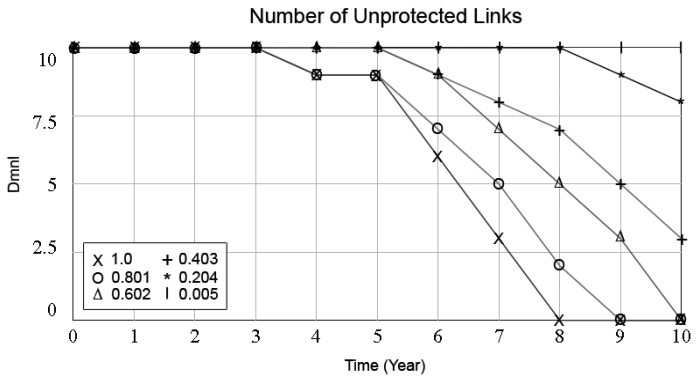


Figure 56: Number of unprotected links

link which won't be considered (practically irrelevant). Therefore, as shown in Fig. 56, even when the uncertainties are very high, the curve with the x's, the number of unprotected links doesn't encounter a sudden decrease. Instead, it is noticed that there is one link that becomes protected at year four then from year four to year five, there are no additional links that become protected, and finally, from year five to year eight there is a constant rate of decrease in the number of unprotected links of value 3 links/year. The reason for this behavior is that at year four the average number of damaged links exceeds the maximum allowable number of damaged links, and in addition, the number of additional links that could be protected had reached a non-fractional value that caused one link to get protected. After that, what happens form year four to year five is that this average drops again below that maximum allowable value, therefore no adjustment is done. Starting from year five, the average goes above the maximum allowable value again, and this time there is a sufficient amount of links that can be protected. This is because the continuation of having a high value for the probability of attack success from the HS's perspective led to the increase in the desired rate of link protection (and the

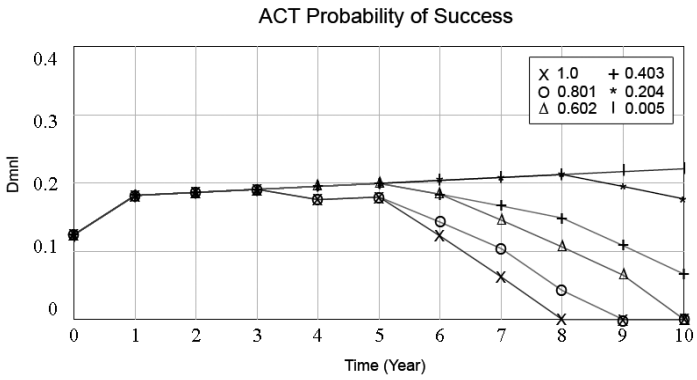


Figure 57: Actual probability of attack success

desire to spend more budget for this purpose. This desire is of course controlled by the tolerable increase in the protection rate).

It can be concluded form the above analysis that it is not necessary to deploy technology all at once. The deployment can be spread over a considerable period of time. This would provide budget relief, the opportunity to deploy new technology advancements, and finally, the chance that circumstances may change and the attacker may get caught.

The ACT probability of attack success is affected by many factors. These factors are: the number of protected and unprotected links, the actual time to discover attack planning and cease attack (this time increases exponentially over a long period of time in this model to account for opposing conditions that may actually act against the HS agency's well), the actual minimum time to plan and commit attack, and other factors (these other factors represent factors that can be added in the future to refine the model. In the current model these factors are represented as a constant equal to 0.1. The complement '0.9' of this constant is multiplied by the remaining factors that affect the actual probability). An

analysis of the simulations results shows that, as shown in Fig. 57, the ACT probability of attack success initially increases from 0.125 to 0.18 during the first year due to two things; the rapid decrease in the ACT time to plan and commit attack, and the slow increase in the ACT time to discover attack planning and cease attack. After one year, the rate of increase of this probability decreases the ACT time to plan and commit attack reaches its minimum possible limit while the ACT rate of discovering attack planning and ceasing attack keeps its slow rate of increase. Starting from year 3, the probability drops with a rate depending on the number of links that get protected which in turn depends on the HS-S probability of attack success in an indirect way.

The analysis of the A-S probability of attack success follows the same direction as the analysis of the ACT probability of attack success. However, there are three differences: First, the A-P number of unprotected links is affected by the protected link obfuscation function which deceives the attacker. Second, the A-P time to discover attack planning and cease attack is also affected by a deception factor that causes time to seem longer to the attacker. Finally, the A-S time to plan and commit attack has a different rate of decrease than the one that affected the ACT probability of attack success. These differences collectively cause the A-S probability of attack success follow the pattern shown in figure 58.

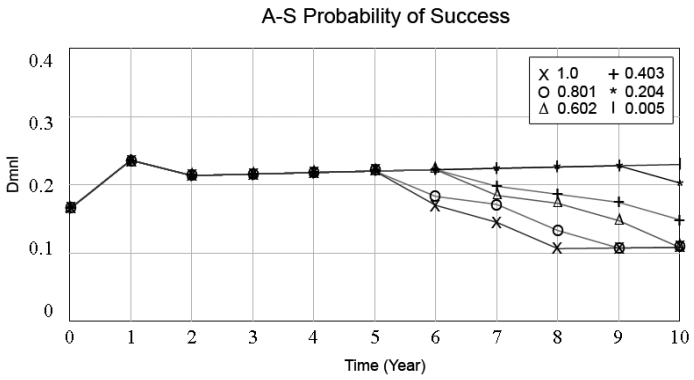


Figure 58: Attacker seen probability of attack success

Case Fixtures	Units
All other variables have values similar to the ones in the previous case $E_K^{HS-S} \rightarrow (we - vary - this)$ $U^{HS-S} = 0.8$	Years / BitsOfInfo Dimensionless

Table 9: Values used for the variables in test case 2

HS-S knowledge effect values used in case study 2			
Value	1.5e − 007	1.16253e − 007	6.75083e − 008

Table 10: HS-S knowledge effect values used for sensitivity analysis

5.1.2. Test Case 2: Varying the Defender-Seen Knowledge Factor

The effect of knowledge on time from HS's perspective is addressed in the second test case. Table 9 lists the variables that have different values from the previous case. The

variable “HS-S Effect of Knowledge on Time” is varied as shown in table 10. Simulation results show that a small change in this variable has an obvious effect on the three probabilities of attack success in the three sub-models. Starting with the first value, 1.5e-007, of the HS-S effect of knowledge on time (the curve with Δ’s in Fig. 59), the HS-S probability of attack success decreases over time starting three years after the simulation start time due to the chosen values for other variables. When the HS-S effect of knowledge on time is decreased, the probability decreases at approximately the same rate it had, but sooner in time. For the values shown in Table 10, the probability decreases starting at year two and one, respectively. For the probability of attack success from the attacker’s perspective (Fig. 60), the general trend of this probability over time is a reduction in its value. The decrease in the effect on knowledge variable causes this decrease to start earlier in time as in the previous case. It is noticed however that there is some non-uniformity in the effect on this probability during first two years. This is, however, not caused by the variable under consideration. It is otherwise due to the effect

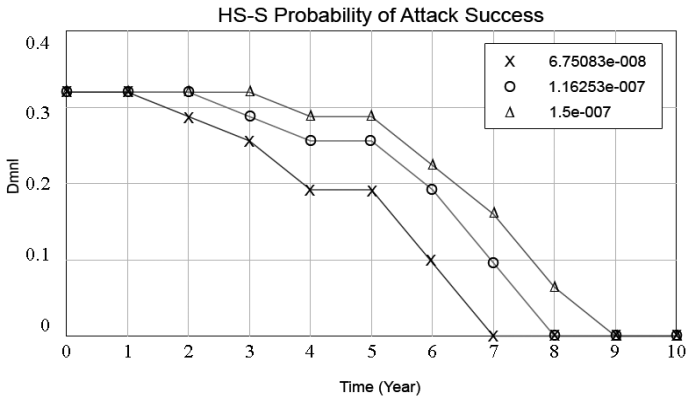


Figure 59: HS-S probability of attack success (test case 2)

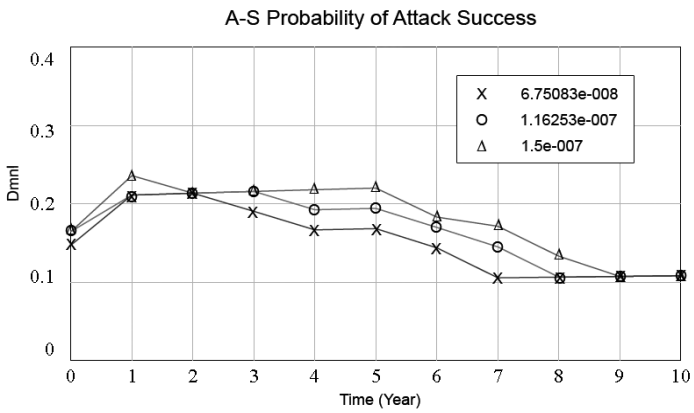


Figure 60: A-S probability of attack success (test case 2)

of the link obfuscation function. The actual probability of attack success (Fig. 61) has the decreasing trend over time like the previous ones but, in general, it has overall smaller values than the other two due to the fact that real world factors that affect this probability may be different from what the two opponents perceive. The effect of decreasing the HS-S effect of knowledge on time on this probability is very similar to its effect on the other two; namely, it makes the decrease start time occur sooner. Again, in the first year it is noticed that this probability increases over time and that no matter what the value of the

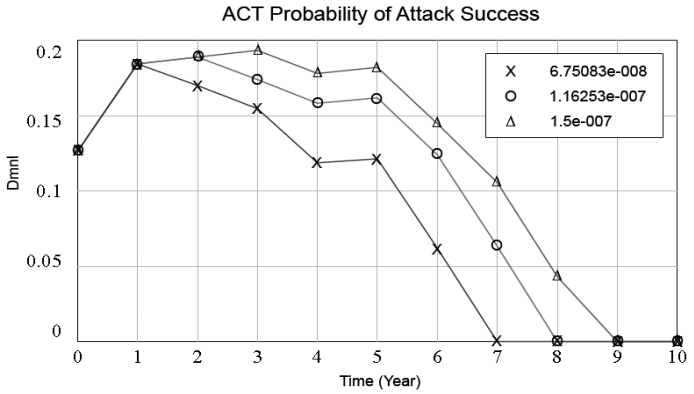


Figure 61: Actual probability of attack success (test case 2)

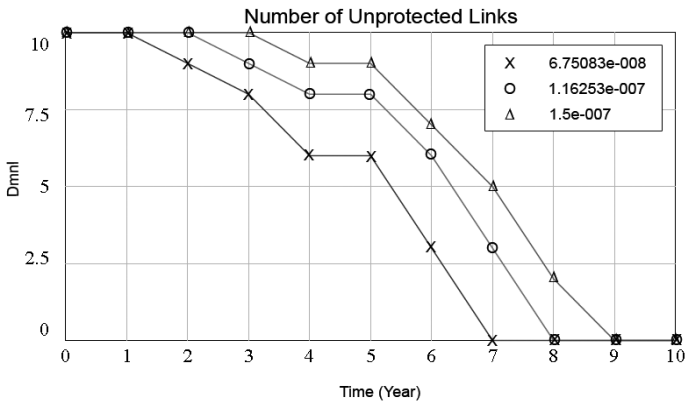


Figure 62: Number of unprotected links (test case 2)

HS-S effect of knowledge on time is the probability is the same. The reason for that is that the number of unprotected links (Fig. 62) is the same during the first year for any value of the effect of knowledge on time and is fixed i.e. doesn't decrease. The above analysis shows that the perception of the HS agency could alone make an observable influence on the actual and the attacker seen probability of attack success. This happens as a result of the decisions made based on that perception that get translated into real world actions (protection deployment) that then affect other system variables. This shows the importance of having a very good perception on the HS's side; this can be supported by very well managed intelligence processes.

5.2. Sensor Deployment

In this section, the Deployment Realization Tool (DRT) is tested on two networks. A sensitivity analysis is conducted on the smoothness factor while fixing the number of deployment radii in the case of the first deployment algorithm. For the second algorithm, the sensitivity analysis is done on charge values and the grid element side length. The

goal is to analyze the resulting deployments and draw some conclusions. The two networks are: the State of Delaware's routes system[1] and Chicago's major streets[2].

5.2.1. Test Case 1: Delaware's Road System

First, we apply deployment method I on the shape file of Delaware's road system. The file contains 70 links representing the main roads. The generated number of BS's for this network is 4 and there is only 1 command center as expected (as we always place the CC at the centroid of the network). These numbers remain fixed for all the tests done on Delaware's network.

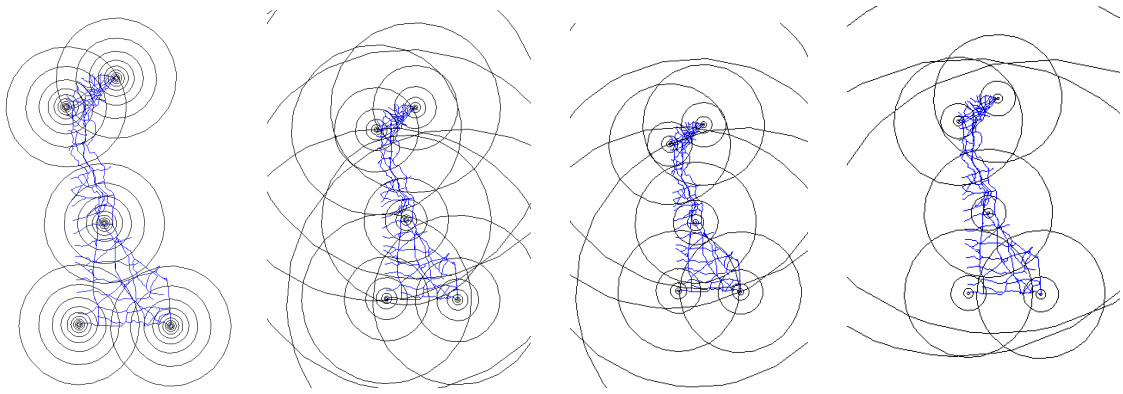
The sensitivity analysis is conducted by varying the value of the deployment smoothness factor F^S and keeping the number of deployment radii fixed at some specific value. Four tests were carried out for this purpose. Table 11 shows the used values of the design parameters. We list the equations again for convenience. The original equations can be found in section 4.2.2.2.1. Fig. 63 shows the deployment radii for each of the tests and Fig. 64 gives the generated sensors. The different values of the smoothness factor are listed in Table 12.

$$N_i = \mathrm{C} \times \exp(-R_i^D / R_m^D)\,;\, R_s^D = 0.5 \times R^T\,;\, R_i^D = R_s^D \times \exp(F^S i) \quad , 0 \le i < F^S$$

Parameter	R^T	R_s^D	C	N_R
Value	2000	1000	1000	10

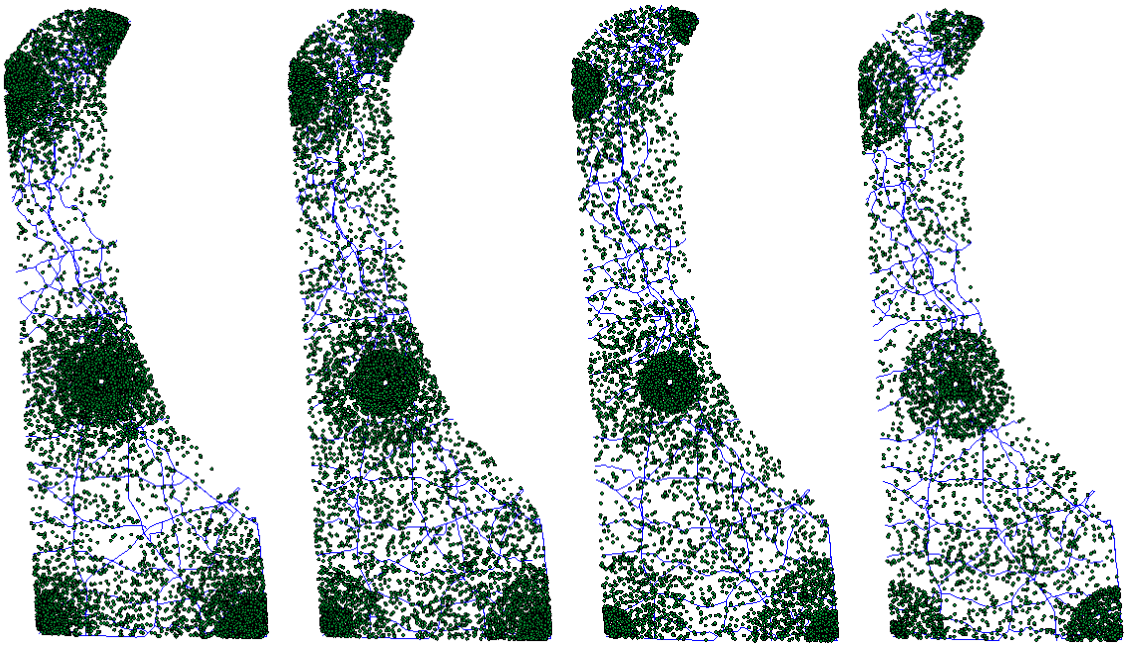
Table 11: Values of the design parameters for the first sensitivity test

[1] The shapefile can be found at - http://www.deldot.gov/information/pubs_forms/gis/centerline/index.shtml
[2] Visit this site for the shapefile - http://data.cityofchicago.org/browse?tags=gis



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

Figure 63: Deployment radii for F^S equals: (a) 0.4 (b) 0.7 (c) 1.0 and (d) 1.3



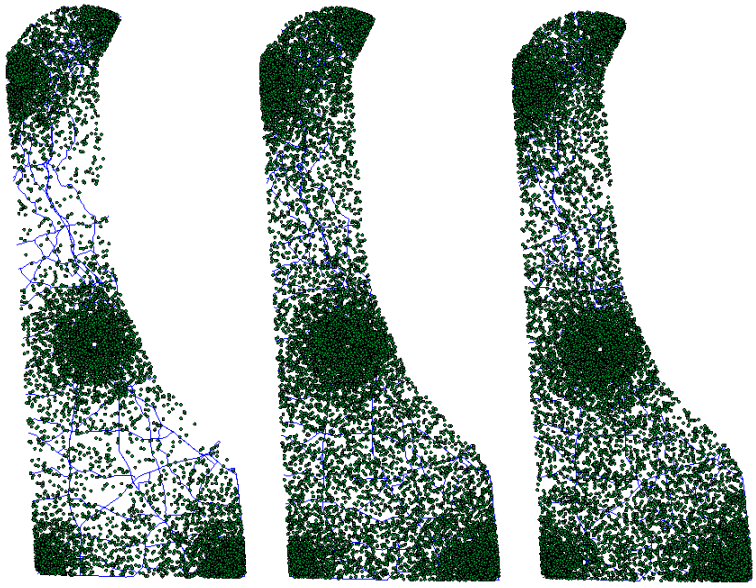
(a) (b) (c) (d)

Figure 64: Generated sensors for F^S equals:(a) 0.4 (b) 0.7 (c) 1.0 and (d) 1.3

Run	F^S	# of sensors
1	0.4	44190
2	0.7	48930
3	1.0	50904
4	1.3	51936

Table 12: Smoothness factor values and the corresponding generated number of sensors

It is easy to see from the above sensitivity tests that to have a smoother deployment, it is better to have a larger number of deployment radii, while having a smaller deployment smoothness factor. This is can be more specifically seen in Fig. 63-a and 64-a. To see this idea more clearly, we fix F^S at 0.4 and increase the number of deployment radii gradually as shown in Fig. 65 and table 13.



(a) (b) (c)

Figure 65: Generated sensors for $F^S = 0.4$ and N_R equals (a) 10 (c) 15 and (d) 30

Run	N_R	# of sensors
1	10	44190
2	15	73884
3	20	103830
4	25	133800
5	30	163770

Table 13: Number of deployment radii and the corresponding generated number of sensors

From the above analysis, is it concluded that a balance between the number of deployment radii and the value of the smoothness factor (which collectively control the actual deployment's difficulty) is necessary to achieve the desired deployment smoothness and consequently reduce traffic congestion near the BS's proportionally. This is an optimization process where the objective is to minimize the number of sensors while maintaining a reasonable smoothness. This problem will be considered in future work.

5.2.2. Test Case 2: Chicago's Major Streets

Chicago's road network was used to test and analyze the sensitivity of the second algorithm to the variation in the charge values and the number of grid elements. Chicago's major streets file that was used contains 359 links. First, the sensitivity of the deployment to the magnitudes of the charges assigned to the BS's, CC, and the links will be studied. This test will be divided into two sections: changing BSs' and CC's charges while fixing those of the highly ranked links, then the reverse. In these two sections, the grid element side length will be fixed to 600. Table 14 shows the values of the charges

allocated to the links based on rank. These values are calculated using a base charge as given in Eq. 45. Table 15 lists the values used for the BSs' and CC's charges and the corresponding total generated number of sensors. Finally, the resulting deployments are shown in Fig. 66.

$$Q_i = Q_b \times (N_{ranks} - i) \tag{45}$$

Where:

Q_b : Base charge

Q_i : Charge per point of link i

N_{ranks} : Number of ranks (= 10 in this work)

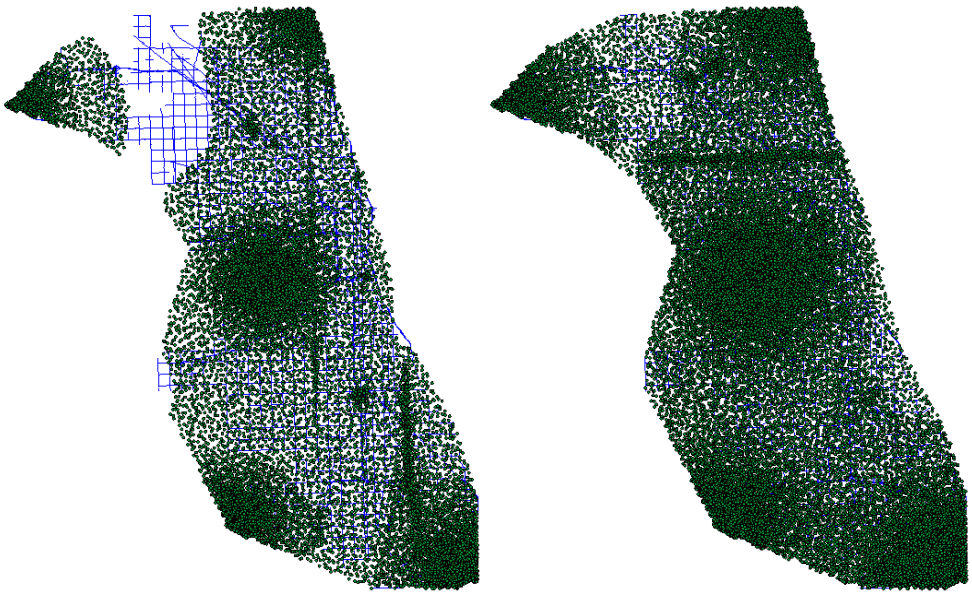
i : Rank

Link Rank	0	1	2	3	4	5	6	7	8	9
Charge (per point)	1.0E-05	9.0E-06	8.0E-06	7.0E-06	6.0E-06	5.0E-06	4.0E-06	3.0E-06	2.0E-06	1.0E-06

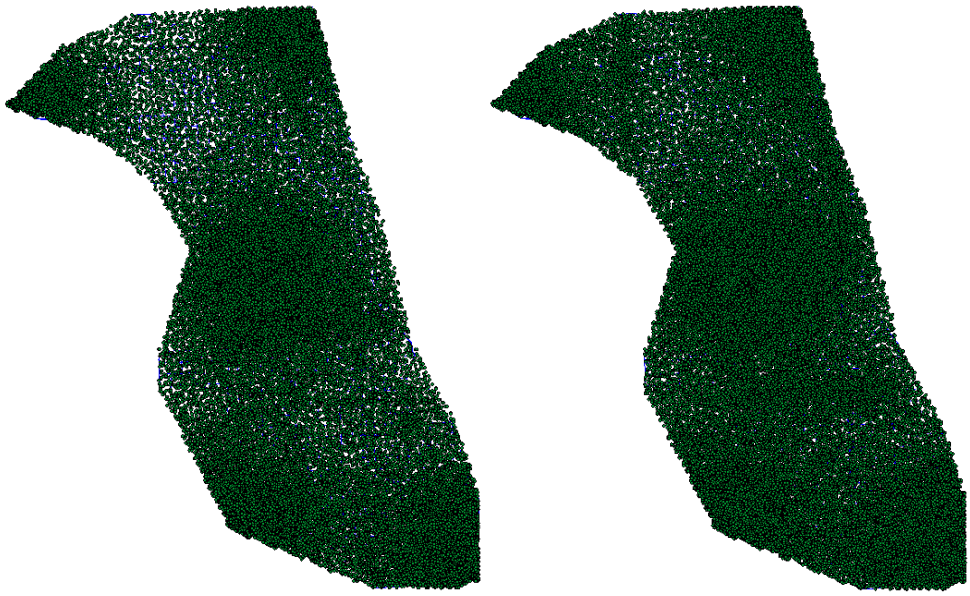
Table 14: Link ranks and the associated charges

BS/CC Charge	0.05	0.1	0.15	0.2
# of Sensors	39501	80534	125265	171671

Table 15: BS/CC charges and numbers of generated sensors



(a) (b)



(c) (d)

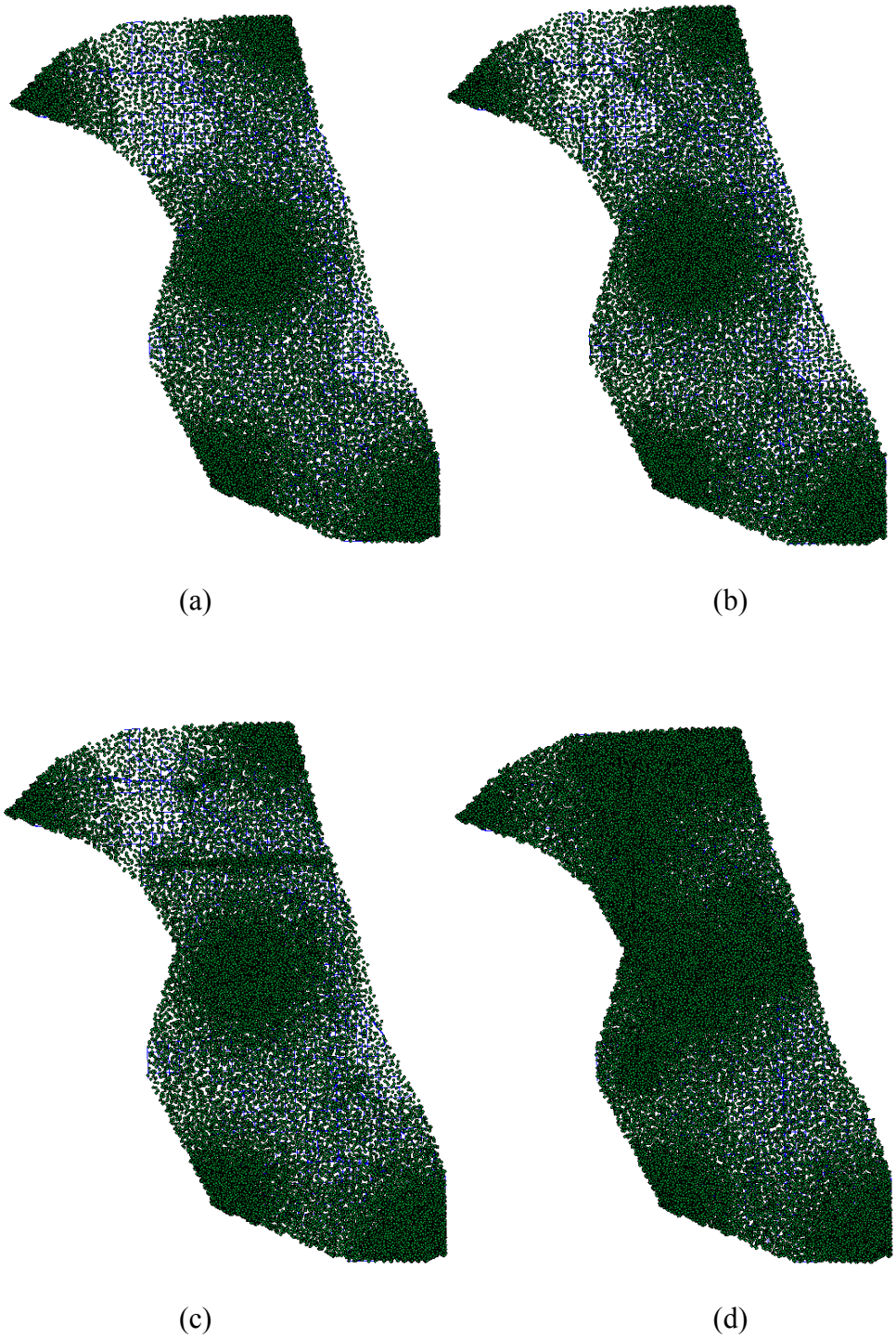
Figure 66: Generated sensors for BS/CC charges equal to a) 0.05 b) 0.1 c) 0.15 d) 0.2.

Note that the ranks were generated randomly and therefore may differ from run to run.

The above tests show that the number of sensors increases linearly with the increase in the charge value. Also, it is clear from Fig. 66-a that for very low charge values, the field can be very weak that it would fail to have any effect on some grid elements, and consequently there will be deployment gaps in the network. This especially happens when the network has some sharp edges that are far from the locations of other BS's or the CC. Therefore, it is necessary to choose an appropriate charge value for the BS's and the CC to assure that their fields intersects. Again, this is an optimization problem as it is required to find the minimum charge that guarantees a gapless deployment. This is due to the fact that the accumulated charge values are mapped to numbers of sensors, i.e. it is required to minimize the number of sensors while keeping a continuous deployment.

In the second section of this test, the base charge value is changed and consequently the links' assigned charges will change. The effect of this change is given in Fig. 67 for a fixed BS's and CC charges of 0.1, and the values of the base charge and the generated sensor numbers are shown in Table 16.

The above test shows that a very little increase in the per-link charge values may lead to a significant increase in the number of sensors. This is expected to greatly affect the overall system cost. Therefore, a careful choice of the value of the base charge is necessary. However, as can be seen in Fig. 67-d, as the values of the charges assigned to the links increase the deployment's nature changes to be more like a regional-importance-based deployment. This means that regions of the city close to highly important links get higher numbers of nodes due to the high field value at these regions. Hence this can be useful if the target is to have regional-importance-based deployment.



(a) (b)

(c) (d)

Figure 67: Generated sensors for base charge equal to (a) 1.0E-08 (b) 1.0E-07 (c) 1.0E-06 (d) 1.0E-07.

Base Charge	1.0E-08	1.0E-07	1.0E-06	1.0E-05
# of Sensors	77230	77870	80534	169671

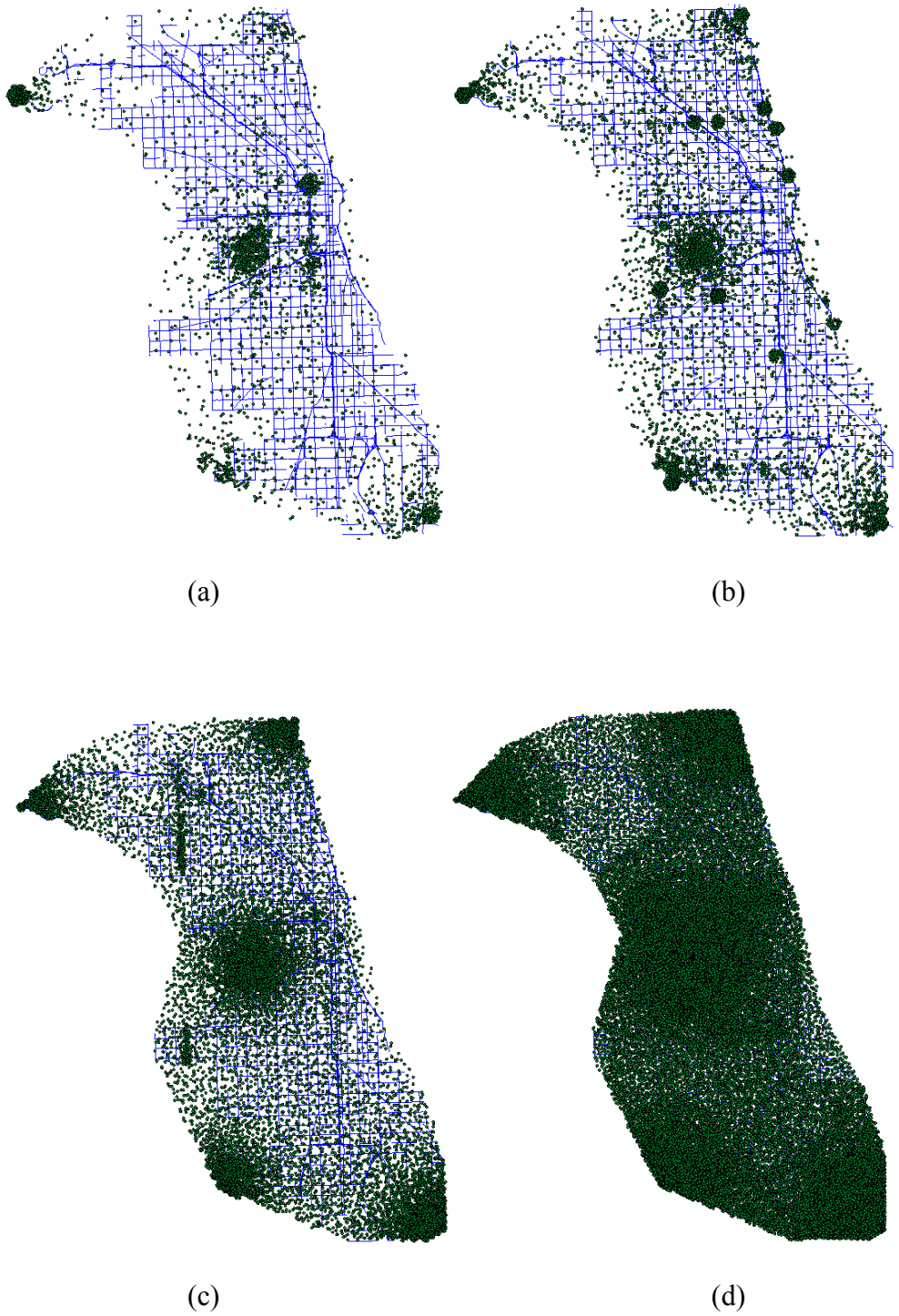
Table 16: Base charge values and numbers of generated sensors

Finally, we end our sensitivity analysis with the grid cell's (hexagon) side length change test. Its effect on the resulting deployment and the number of sensors is studied. For that purpose, we fix both the links' base charge at 1.0E-06 and the BS/CCs' charges at 0.1, and only vary the grid element's side length. Similar to the above analyses, the values of the grid elements' side length values are provided in Table 17 along with the resulting number of sensors, and the deployments are shown in Fig. 68.

GE's Side Length	3000	2000	1000	500
# of Sensors	2736	9008	29772	211294

Table 17: Grid element's side length values and numbers of generated sensors

The deployment figures indicate that as the side length of the grid cell decreases, the accuracy of the deployment increases, as expected. This is because the number of grid cells increases, the field is evaluated at more and more points (grid cell centers), and the more these points are the more accurate the calculated densities of the sensors placed in each cell. One can think of this as if the cell is approaching a zero size; if that happens, the field value would be found at every point in the network's area.



(a) (b)

(c) (d)

Figure 68: Generated sensors for grid element side length equal to (a) 3000 (b) 2000 (c) 1000 (d) 500.

There are two concerns, however, about increasing the number of grid cells: First, the computational complexity becomes high very fast, and the second is that the number of generated sensors increases nearly exponentially. This leads us to the same result that we got previously: This is an optimization problem, where it is required to find the optimal number of grid cells that achieve the highest deployment accuracy within a budget constraint, for example.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

6.1. Summary

In this work, one modeling and two algorithmic contributions were proposed along with two homeland security assistive tools. The first contribution is a model which describes the interaction between an attacker and a homeland security agency and prescribes a plan of actions for the HS agency. The second is the proposal of two WSN deployment algorithms. Regarding the tools, the first tool is intended to serve as a decision support tool for technology deployment plans. It takes measurable input values of different variables extracted from intelligence information and produces a set of assistive charts that guide the deployment plans. The produced charts provide the HS agency with indicators about how far the collected intelligence information may be from the reality, the vulnerability of the network, and the time frame over which the agency can deploy technology while maintaining an acceptable level of safety. This tool was implemented both in Java and in VenSim, but it does not have a GUI currently. The second tool is a sensor deployment tool, which has been implemented in Java and has a ready to use GUI. It accepts a GIS-data ‘shapefile’ that contains the roads of the network under consideration. The tool enables a user to select from two sensor deployment strategies and produces the corresponding deployments in the form of shapefiles containing node distributions and a textual output that contains the information needed for the actual deployment.

The two tools were tested on several inputs and a sensitivity analysis has been done for some test cases. The results of the first tool indicate that technology deployment need not be carried out all at once for the network under consideration while being able to maintain a desired safety level decided by the HS agency. It was also shown that the probability of attack success from the perspectives of the attacker and the HS agency may differ significantly from one another and from the actual probability, and that the perception of the HS agency alone can have a significant impact on the results of the interaction. As for the second tool, it was shown that the second deployment method is better than the first. This is because it takes link ranks into consideration, and is expected to distribute the concentrated effect of high traffic near the BSs, which will highly likely form big holes around the BSs, over the whole network with a very small granularity that in turn weakens that effect. The ability of this tool was also tested through the generation of the two deployments for two networks changing different design parameters for each and testing the results.

6.2. Usefulness of the Results

The usefulness of the proposed attacker-defender interaction model and the accompanying tool will depends mainly on the availability of intelligence data. Great effort was exerted to ensure that the variables used in the model represent measurable quantities that are accessible to intelligence agencies. We believe that coordination between information collecting agencies can serve this goal greatly. Once the inputs to that model are available, its use is straightforward.

As discussed in Chapter 4, the model produces a set of supportive graphs that represent the probabilities of attack success from different perspectives and a technology

deployment time plan represented in the rate of link protection graph. It is believed that the availability of such set of outputs significantly serves the process of assessing the vulnerability of the system, analyzing the risk of being attacked, and allocating the monetary expenses needed for system protection.

All the outputs are given in a per-year form. This should be helpful in studying the contributions of different system factors to the possibility of having a successful attack at a specific year, for example. It also gives the overall trend of that probability which is useful to make future predictions. The availability of three probabilities of attack success, two of them describing the perceptions of the two opponents and the third representing the 'almost' actual probability, is useful for the HS agency in determining its need for track corrections and strategy change decisions. At the end, the model is a self-assessment and decision support tool that is expected to help HS agencies to make better decisions.

In the design of the deployment approaches presented in this work, two considerations were given attention; addressing WSNs' design issues and producing an easy to use deployment tool that gives an easy to use output. The first proposed algorithm produces a set of deployment radii/contours and an accompanying set of sensor densities for each inter-radius ring. The second produces a set of hexagonal grid cells' center locations, cell side length, and the density of sensors per cell. These outputs are sufficient to describe the respective deployments completely as the deployment of individual sensors is random. The sample outputs presented throughout the work show that using the generated outputs can provide smooth deployments that address different WSN related

issues. Also, the use of GIS data gives a realistic view of how the deployment will be and the obstacles that may be encountered in the physical deployment.

In conclusion, it is thought that the outputs of the proposed models and approaches are expected to be useful for the purposes of self assessment, decision support, deployment planning and decisions, and in the actual deployment.

6.3. Practicability of the Results

The practicability of the HS-attacker model and tool is not a concern as they provide graphical and numerical outputs used for assessment and decision support. The major concern would be with the deployment approaches provided. A natural question that may be asked is: Are these deployment approaches realistic and practicable? The answer to this question has two partitions, one related to the feasibility of the approaches and the other is policy and public acceptance related. Starting with the first, the sensor numbers generated in the simulations (bearing in mind the large scale of the deployment) show that an appropriate choice of the design parameters can produce very reasonable numbers of sensors from a practical viewpoint. In Addition, the outputs, represented in the deployment radii and the grid cell hexagons can be easily used through the traditional surveying techniques.

The other concern is public acceptance of these deployments and the associated policies. It is clear that the proposed approaches provide a deployment that requires a city-wide coverage. Spreading these large amounts of sensors over the whole city necessitates that sensors should be deployed within the boundaries of numerous private properties. This is expected to be considered by many as a privacy violation issue. Even though the goal is to provide public safety, different perceptions of the existence of such

devices in the surroundings will definitely be diverse. Whether policies that mandate the placement of these devices should be created or not is another issue. Therefore, careful studies of the possible solutions to this concern are required.

6.4. Conclusions

Many conclusions can be drawn from the proposed models and approaches. In the context of the interaction between homeland security agencies and attackers, it has been concluded that the difference in perceptions due to the different views of each opponent of the real world and the other, can lead to significantly different perceptions of the probability of a successful attack. Another important conclusion is that the perception of the HS agency has its important effect on the interaction. Therefore, HS agencies should work hard to reduce the uncertainties of their knowledge about the attacker. Additionally, what represents a chance to the attacker should be investigated deeply, as these chances are a basic hidden player in the interaction. Finally, it was found that the technology deployment process can be spread over some time period while maintaining the safety level desired by the HS agency.

Deployment-wise, it has been concluded that networks with sharp corners can cause deployment gaps if insufficient charge values were used at the base stations and the command center. Also, the grid cell elements, used to find the resulting field at different network points, were found to produce better results as their size decreases. This happens as the decreased size allows for field evaluation at more points and consequently produces more accurate sensor densities at different locations.

6.5. Future Work

Several areas of enhancement exist for the current work. Regarding the attacker-defender model, it is intended to discover methods for model validation and sub-model refinement. It is also very important to explore methods for modeling "attacker chances". Chances play a very important role in the model; they participate in the attacker's decision process of whether or not to attack. When a chance exceeds a pre-specified threshold, it becomes active. In the current model, chances were generated randomly with a uniform distribution. However, finding the correct distribution by taking into consideration the readiness of the attacker at the time of the chance, the history information of the attacker (which decides the suitability of the chance and its correlation to other chances), and the state of the targeted link (protected/unprotected), is necessary for correct chance modeling.

Other very important aspects to develop include: the consideration of different attacker types, different target system elements, and the non attack alternative on the attacker's side.

Improvements will be done to the sensor deployment algorithms by optimizing the number of sensors while avoiding deployment gaps and maximizing the data traffic flow. Trials will be made to restrict the deployment to network links only taking into consideration the congestion issue. Additionally, a data routing protocol will be developed, according the specification mentioned in Chapter 2, and tested on the proposed deployments for end-to-end delay, delivery ratio, etc.

Finally, more features will be added to the already existing tools, such as WSN routing support, and development will continue for the uncompleted ones.

REFERENCES

A. Miu, “Lecture 7: Voronoi Diagrams”, Computational Geometry course slides, 2001, [Online]. Available: http://nms.csail.mit.edu/~aklmiu/6.838/L7.pdf

Amenaza Technologies Limited, "Creating Secure Systems through Attack Tree Modeling", Report, Calgary, AB, Canada, 2003

B. Jackson, P. Chalk, R. Cragin, B. Newsome, J. Parachini, W. Rosenau, E. Simpson, M. Sisson, and D. Temple, “Breaching the Fortress Wall Understanding Terrorist Efforts to Overcome Defensive Technologies,” Homeland Security Program within RAND Infrastructure, Safety, and Environment, Santa Monica, CA, Tech. Rep., 2007.

B. Ezell, S. Bennett, D. Winterfeldt, J. Sokolowski, and A. Collins, "Probabilistic Risk Analysis and Terrorism Risk", Risk Analysis, Vol. 30, No. 4, 2010

B. Guru and H. Hiziroğlu, “Electromagnetic Field Theory Fundamentals,” PWS Publishing Company, 1998.

C. Kirkwood. System Dynamics Methods: A Quick Introduction, 2010, [Online]. Available: http://www.public.asu.edu/~kirkwood/sysdyn/SDIntro/SDIntro.htm

C. Wenjie, G. Liqiang, C.i Zhilei, C. Zhanglong, and T. Shiliang; “An Intelligent Guiding and Controlling System for Transportation Network Based on Wireless Sensor Network Technology,” Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on, 2005 , Page(s): 810 - 814

C. Westen, “Introduction to Risk Assessment”, Presentation, Department of Geography, Makerere University, 12-23 September, 2005

D. Dornan and M. Maier, “Incorporating Security into the Transportation Planning Process,” National Cooperative Highway Research Program Report 525, Surface Transportation Security, Volume 3, Transportation Research Board, Washington, D.C., 2005

D. Douglas & T. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," The Canadian Cartographer 10(2), 112–122 (1973)

E. Jeneliusa, J. Westina, and A. Holmgren, “Critical Infrastructure Protection Under Imperfect Attacker Perception,” International Journal of Critical Infrastructure Protection, vol. 3, no. 2010, pp. 16–26, 2010.

E. Paté-Cornell and S. Guikema, “Probabilistic Modeling of Terrorist Threats: A Systems Analysis Approach to Setting Priorities Among Countermeasures,” Military Operations Research, Vol. 7, No 4, pp. 5-20 December 2002

Florida Department of Transportations District 6: Intelligent Transportations Systems, [Online].Available:http://www.sunguide.org/sunguide/index.php/services/services/intelligent_transportation_systems/70/its_devices/cctv

G. Ackerman, P. Abhayaratne, J. Bale, A. Bhattacharjee, C. Blair, L. Hansell, A. Jayne, M. Kosal, S. Lucas, K. Moran, L. Seroki, and S. Vadlamudi, “Assessing Terrorist Motivations for Attacking Critical Infrastructure,” Lawrence Livermore National Laboratory, University of California, Technical 227068, Jan 2007.

G. Richardson and P. Otto, “Applications of System Dynamics in Marketing: Editorial,” Journal of Business Research, vol. 61, no. 11, pp. 1099–1101, Nov 2008.

G. Parnell, C. Smith, F. Moxley ,“Intelligent Adversary Risk Analysis: A Bioterrorism Risk Management Model ,“ Submitted to Risk Analysis, Journal of the Society for Risk Analysis, February 20, 2009

H. Blum, “A Transformation for Extracting New Descriptors of Shape,” In Models for the Perceptions of Speech and Visual Form, Weiant Wathen-Dunn (ed.), MIT Press, Cambridge, MA

H. Willis, A. Morral, T. Kelly, J. Medby, "Estimating Terrorism Risk", Report, Center for Terrorism Risk Management Policy, RAND Corporation, 2005

H. Tsang, A. Park, M. Sun, and U. Glässer, “GENIUS: A Computational Modeling Framework for Counter-Terrorism Planning and Response,” ISI 2010, May 23-26, 2010, pp. 71 - 76 , Vancouver, BC, Canada

I. Insua, J. Rios, and D. Banks, “Adversarial Risk Analysis,”Journal of the American Statistical Association, June 2009, Vol. 104, No. 486

J. Homer and G. Hirsch, “System Dynamics Modeling for Public Health: Background and Opportunities,” American Journal of Public Health, vol. 96, no. 3, pp. 452–458, Mar 2006.

J. Rios, “Adversarial Risk Analysis for Counter Terrorism Modeling,”Workshop on Adversarial Decsion Making, DIMACS, September 2010

J. Sterman, “System Dynamic Modeling for Project Management,” System Dynamics Group, Massachusetts Institute of Technology, Cambridge, MA, Tech. Rep., 1992.

J. Victoroff, “The Mind of the Terrorist: A Review and Critique of Psychological Approaches,” The Journal of Conflict Resolution, vol. 49, no. 1, pp. 3–42, Feb 2005.

J. Almasizadeh, M. Azgomi, “A New Method for Modeling and Evaluation of the Probability of Attacker Success,“ 2008 International Conference on Security Technology

K. Al-Khateeb and J. Johari, “Intelligent dynamic traffic light sequence using RFID,” Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on , vol., no., pp.1367-1372, 13-15 May 2008

L. Yuan; Y. Zhu, "Modeling and Simulating Wireless Sensor Transportation Monitoring Network," Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on , vol.2, no., pp.8640-8644

L. Mimbela, L. Klein, P. Kent, J. Hamrick, K. Luces, and S. Herrera, “A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems”, handbook, August, 2007

M. Bell, U. Kanturska, J. Schmocker, and A. Fonzone, "Attacker–defender models and road network vulnerability,"Philosophical Transactions of the Loyal Society, Vol. A (2008) no. 366, pp. 1893–1906 Published online 6 March 2008

M. Crenshaw, "The Psychology of Terrorism: An Agenda for the 21st Century," Political Psychology, International Society of Political Psychology, vol. 21, no. 2, pp. 405–420, Jun 2000.

M. Cummings, D. McGarvey, P. Vinch, "Homeland Security Risk Assessment - Methods, Techniques, and Tools," Volume II. , report , June 16, 2006, Homeland Security Institute

M. Franceschinis, L. Gioanola, M. Messere, R. Tomasi, M. Spirito, and P. Civera, "Wireless Sensor Networks for Intelligent Transportation Systems," Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th , vol., no., pp.1-5, 26-29 April 2009

M. Radzicki and R. Taylor. (1997) U.S. Department of Energy's Introduction to System Dynamics: A Systems Approach to Understanding Complex Policy Issues. [Online]. Available: http://www.systemdynamics.org/DL-IntroSysDyn/

M. Rausand, "System Reliability Theory; Models, Statistical Methods and Applications", Book, (Second Edition), Wiley, 2004

N. Howard, "Application of Deterrence Assessment Methodologies to Commercial Ferry Attack Scenario," in IEEE Conference on Technologies for Homeland Security, Boston, MA, 2005.

Intelligent Transportation Society of America in Cooperation with the United States Department of Transportation, "Homeland Security and ITS: Using Intelligent Transportation Systems to Improve and Support Homeland Security," Supplement to the National ITS Program Plan: A Ten-Year Vision, September 2002

Q. Wang, L. Fiondella, N. Lownes, J. Ivan, R. Ammar, S. Rajasekaran, and S. Tolba. Integrating Equilibrium Assignment in Game-theoretic Approach to Measure Many-to-Many Transportation Network Vulnerability. In Proc. of 11th IEEE International Conference on Technologies for Homeland Security, Waltham, MA, Nov 2011.

R. Borum, "Psychology of Terrorism." Tampa: University of South Florida, 2004.

R. Hudson, "The Sociology and Psychology of Terrorism: Who Becomes a Terrorist and Why?" Federal Research Division, Library of Congress, Tech. Rep., Sep 1999, prepared under an Interagency Agreement.

R. Gonzalez, R. Woods, "Digital Image Processing," Third Edition, Pearson Prentice Hall, 2008, p.p. 651-654, 812-815

R. Sedgewick and K. Wayne, "Geometric Algorithms," Presentation, Algorithms and Data Structures course slides, Princeton University, 2007

S. Chapra and R. Canale, "Numerical Methods for Engineers," Fifth Edition, McGraw-Hill, 2006, p.p. 332

S. Cheung and P. Varaiya, “Traffic Surveillance by Wireless Sensor Networks: Final Report,” California PATH Research Report, California Path Program, Institute of Transportation Studies, University of California, Berkeley, January 200

S. Toumpis, L. Tassiulas, “Optimal deployment of large wireless sensor networks,” Information Theory, IEEE Transactions on , vol.52, no.7, pp. 2935- 2953, July 2006

S. Boyd and L. Vandenberghe, “Convex Optimization,” Cambridge University Press, 2004, p.p. 23-24

S. Fortune, “A sweepline algorithm for Voronoi diagrams,” Proceedings of the second annual symposium on Computational geometry. Yorktown Heights, New York, United States, pp.313–322. 1986.

S. Tolba, L. Fiondella, R. Ammar, N. Lownes, S. Rajasekaran, J. Ivan, and Q. Wang, “Modeling Attacker-Technology System Interaction in Transportation Networks: P2I3-Model,” In Proc. of 11th IEEE International Conference on Technologies for Homeland Security, Waltham, MA, Nov 2011

W. Green, “Hazard, Threats, Risk, Etc.: An examination of some key terms …”, Presentation, Disaster Theory Series No. 3, School of Continuing Studies, University of Richmond, 2008

Wikipedia the Free Encyclopedia, Online: http://en.wikipedia.org/wiki/Centroid

Y. Chen; L. Cheng; C. Chen; J. Ma; , "Wireless Sensor Network for Data Sensing in Intelligent Transportation System," Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th , vol., no., pp.1-5, 26-29 April 2009

VITA

Sherif Ahmed Tolba received his Bachelor of Science degree in electronics and communications from Mansoura University in Egypt 2007. He entered the Computer Science and Engineering program at The University of Connecticut in September 2009. His research interests include routing protocols in wireless sensor networks, sensor deployment, and risk analysis. He plans to proceed to the Ph.D. program at the CSE department, University of Connecticut.

Mr. Tolba may be reached at 365 South Street, Willimantic, 06226, USA. His email is sherif.tolba@engr.uconn.edu.