Evaluation and Analysis of Node Localization Power Cost in Ad-hoc Wireless Sensor Networks with Mobility

THESIS

Brian A. Sessler, First Lieutenant, USAF

AFIT/GCE/ENG/06-07

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCE/ENG/06-07

Evaluation and Analysis of Node Localization Power Cost in Ad-hoc Wireless Sensor Networks with Mobility

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Engineering

Brian A. Sessler, BS

First Lieutenant, USAF

March 2006

AFIT/GCE/ENG/06-07

Evaluation and Analysis of Node Localization Power Cost in Ad-hoc Wireless Sensor Networks with Mobility

Brian A. Sessler, BS
First Lieutenant, USAF

Approved:

/signed/	6 Mar 2006
Dr. Rusty O. Baldwin (Chairman)	date
/signed/	6 Mar 2006
Dr. Barry E. Mullins (Member)	date
/signed/	6 Mar 2006
Dr. John F. Raquet (Member)	date

AFIT/GCE/ENG/06-07

Abstract

This research examines the power costs of network irregularities on communications and localization in an Ad-hoc Wireless Sensor Network (AWSN). Specifically, the performance of two *anchor*-based algorithms, APS-Euclidean and Map-Growing, are characterized with respect to the data communications costs. The number of data bits transmitted and received are significantly affected by varying levels of mobility, node degree, and network shape. For APS-Euclidean, mobility accounts for 92.91% of the variation in the number of bits transmitted. The highest level of mobility resulted in 672% more transmitted bits than the corresponding static network. The concurrent localization approach, used by the APS-Euclidean algorithm, has significantly more accurate position estimates with a higher percentage of nodes localized, while requiring 50% less data communications overhead than the Map-Growing algorithm. Analytical power models capable of estimating the power required to localize are derived. The average amount of data communications required by either these algorithms in a highly mobile network with a relatively high degree, consumes less than 2.0% of the power capacity of an average 560mA-hr battery. This is less than expected and contrary to the common perception that localization algorithms consume a significant amount of a node's power.

The potential of an AWSN to be self-organizing, scalable, and fault-tolerant makes it a very promising Command, Control, Communications, and Information (C3I) tool for the military to perform surveillance, reconnaissance, and target tracking missions. One of the key concerns with location aware AWSNs is how sensor nodes determine their position. Although position estimation accuracy is a primary goal of localization, it is also necessary to minimize the power consumption of the process. The inherent power limitations of an AWSN along with the requirement for long network lifetimes, makes achieving fast and power-efficient localization vital.

Acknowledgements

I would first like to recognize and thank my wife and children for their continuous love, support, and sacrifice during my studies. I also owe a special thanks to my mother and father, for the support and encouragement they have given me throughout my life.

I owe a great thanks to my thesis advisor, Dr. Rusty Baldwin, whose guidance and encouragement was extremely helpful throughout this research effort. Specifically, Dr. Baldwin's instructions on research and performance analysis techniques were key in the successful completion of this thesis. I would also like to extend appreciation to my committee members, Dr Barry Mullins and Dr John Raquet, for their guidance and support.

Brian A. Sessler

Table of Contents

Page

List of Figures

List of Tables

List of Abbreviations

Evaluation and Analysis of Node Localization Power Cost in Ad-hoc Wireless Sensor Networks with Mobility

I. Introduction

1.1 Motivation

Ad-hoc Wireless Sensor Networks (AWSNs) can be used in many military applications. A self-organizing, scalable, and fault-tolerant AWSN would be a very promising command, control, communications, and information (C3I) tool for the military to perform surveillance, reconnaissance, and target tracking missions. AWSNs could be used to passively and secretively monitor, detect, and identify the presence or movement of enemy forces, terrorists, or even weapons of mass destruction in areas where the presence of ground forces is not feasible. Additionally, the integration of AWSNs into military "Blue Force Tracking" systems can contribute to mission success while avoiding losses due to friendly fire. Both of these systems will require accurate, reliable, and low cost methods of achieving and maintaining sensor location, while maximizing the lifetime of the network.

1.2 Background

AWSN technology is still in its infancy, and there is much work to be done before we tap their full potential. For example, achieving reliable, accurate, and energy-efficient node localization is an immature area. Node localization is the process of deriving physical or virtual coordinates of nodes in a sensor network. It is crucial for reporting the location of an event for location-aware applications, evaluating network coverage, and assisting with optimal-path multi-hop routing to minimize power consumption. AWSNs by definition are unable to deploy with a priori knowledge of node placement. Additionally, allowing every node to self-localize with Global

Positioning System (GPS) receivers is costly in terms of dollars, size, and power consumption. Thus, many AWSN localization algorithms rely on a small percentage of nodes equipped with GPS to serve as *anchors* and assist in a distributed localization algorithm. Although there are also several *anchor*-free localization algorithms that provide relative or virtual location estimates, this research is limited to *anchor*-based algorithms, where every node in the network belongs to one of two categories, *anchors* and *unknowns*. *Anchors* are able to self-localize via GPS or some other means. *Unknowns* require the help of *anchors* and possibly other unknowns to estimate their positions.

Several network and environmental factors impact the performance of node localization algorithms. For instance, network shape irregularities (due to geographic/environmental conditions), varying network topologies (network connectivity and node distributions) and node mobility introduce many challenges to achieving localization. Several localization algorithms, such as any where position estimates are based on shortest-path distances, will simply not perform well in networks with irregular network shape. Furthermore, many algorithms may also have thresholds on the minimum average node degree necessary for the network to successfully localize or converge. Non-uniformity of node placement typically has a negative effect on the accuracy and success of a localization algorithm. Node mobility arguably introduces the most perplexing challenge, as initially achieving and maintaining up-to-date position estimations for mobile nodes ultimately requires additional, and possibly continuous, localization traffic.

1.3 Research Objectives & Hypothesis

A major objective of localization within AWSNs is to minimize the power consumption of the process. The inherent power limitations of an AWSN along with the requirement for long network lifetimes, makes achieving fast, power-efficient, and accurate localization vital to the success of an AWSN application. Operating for extended periods with limited battery power is a major concern and challenge. The

main goal of this research is to determine the effect of network shape, node degree, and node mobility on the power required to localize nodes in an AWSN. The hypothesis of this research is that the more irregular or dynamic the network configuration, the more power required to achieve localization.

Another objective of this research is to determine the performance differences between a incremental and a concurrent localization algorithm in irregular networks with node mobility. For this, the incremental algorithm Map-Growing [LSS04] and the concurrent algorithm APS-Euclidean [NN01] are used as representative of the respective approaches. Although the research examines the effects of these factors on convergence and accuracy, the main focus is to determine the data communications cost of the two algorithms. Given that Map-Growing is a multi-phase incremental algorithm, it is hypothesized that it will have a higher communications overhead than the concurrent APS-Euclidean algorithm.

1.4 Approach

Simulation models of the Map-Growing and APS-Euclidean algorithms are developed and observed under various experimental configurations. Mean position error, percent localized, bits transmitted, and bits received are all performance metrics of interest. Next, the effects of the various levels of node mobility, degree, and network shape are computed and analyzed. The results are analyzed to determine the significance of the effect on the response for each factor level. Additionally, an analysis of variance is performed to determine the factors or interactions that explain the highest percentage of variation in the response. The secondary objective of this research is accomplished by simultaneously analyzing the performance differences and similarities of the two algorithms.

Additionally, a multiple linear regression on the Bits Transmitted and Bits Received responses is performed, in order to derive a power model for estimating power consumption given different factor levels. The results of the data communications

overhead responses and the associated power costs are further analyzed with respect to percentage of battery capacity consumed.

1.5 Summary

The capabilities of AWNSs in passive surveillance and target tracking applications make it a promising technology for future Air Force and other Department of Defense weapon systems. One of the key concerns of AWSNs is achieving accurate node localization while not significantly reducing the expected lifetime of the network. The primary focus of this research is to examine the data communications cost and associated power requirements of performing node localization in networks with node mobility and varying network degree and shape.

This document is organized as follows. Chapter 2 contains the background information and associated challenges of node localization in wireless sensor networks. Chapter 3 describes the methodology used to evaluate the effects of networks shape, network degree, and node mobility on AWSN localization. Chapter 4 presents the experimental results and analysis. Lastly, Chapter 5 summarizes the conclusions of the research and discusses its significance.

II. Background

2.1 Introduction

This chapter presents an overview of the fundamentals of localization and explores challenges associated with achieving "ideal" localization in Wireless Sensor Networks. Section 2.2 describes some of the characteristics, constraints, and difficulties associated with achieving accurate and efficient node localization. Section 2.3 introduces the fundamentals of several different ranging methods often used in AWSNs. Section 2.4 explains the process of propagating ranging and position information and introduces various techniques for doing so. Section 2.5 explores several basic methods for computing position estimation such as the most common method, trilateration. Section 2.6 describes how to improve initial node position estimates through a handful of different iterative refinement methods. Finally, Section 2.7 concludes this chapter looking at relevant research dealing with node localization in anisotropic/irregular and mobile networks.

2.2 Fundamentals of Localization

Localization is the process by which sensor nodes deduce their physical, global, or relative position in a sensor network. Node localization in AWSN applications has been an active area of research in recent years, with a focus towards achieving accurate, efficient and cost effective localization of ad-hoc networks. Detailed surveys of such systems can be found in [ASSC02] and [HB01b]. Self-organizing, scalable, fault-tolerant, and energy-efficient are four important design characteristics of a typical AWSN. Unfortunately, these four characteristics, along with the requirement for a low-cost and micro-size node platform, present formidable constraints and complications that ultimately impact how node localization is accomplished. The remainder of this section discusses the impact these characteristics have on localization, presents a general localization structure, and discusses some of the obstacles associated with common ranging and position estimation approaches.

The self-organizing characteristic implies an AWSN operates independently. Thus, every node in the network must determine its location without a priori of coordinates or obtaining them through a fixed infrastructure. From this single requirement, the need for a scalable, fault-tolerant, and energy-efficient method of localization arises. Localization algorithms can be classified by their accuracy (i.e., coarse-grained versus fine-grained localization), by whether or not they use distances to derive position estimates (i.e., range-based versus range-free), by whether they allow nodes to localize concurrently or incrementally, and lastly by whether they are implemented in a distributed or a centralized manner.

Localization is typically achieved using either a distributed or a centralized algorithm. The centralized algorithm collects range measurements and node locations at a central node, where the localization computations take place and the resulting position estimations are sent back to the respective *unknowns*. The distributed localization algorithm has each node estimate its own location until accuracy requirements for position estimation are met. In wireless sensor network applications where capable hardware and power supplies are available, centralized localization algorithms can be very accurate, efficient, and effective. Multi-Dimensional Scaling (MDS) algorithms have been used to perform centralized localization (MDS-MAP) in an AWSN [SPS02], since the algorithm requires the all-pairs shortest path distance between all of the nodes in the network to accurately estimate positions. However, given a resource constrained AWSN and the unpredictability of the deployed environment, a centralized algorithm introduces several problems. The most crucial is it creates a single point-of-failure; if the central node is not appropriately positioned or it fails, the network will not be able to localize. Centralized approaches also require support mechanisms such as leader election and an efficient non-location based routing protocol which results in additional communication overhead. For example, Figure 2.1 shows a centralized implementation of the Ad-Hoc Localization System (AHLos) with six to ten times more communications overhead than a comparable distributed implementation [SHS01].

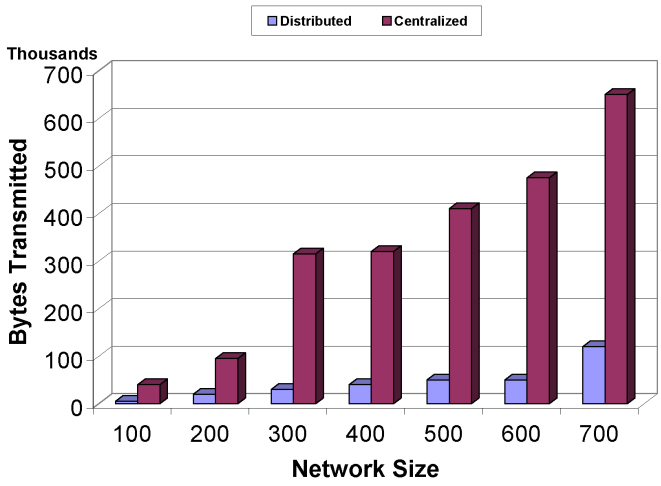

Figure 2.1: Network Traffic in Distributed versus Centralized AHLoS Algorithms [SHS01]

Another major problem with centralized localization is nodes close to and around the central node route more traffic and thus use more power; ultimately leading to a shorter life-expectancy of the heavily used nodes and therefore the entire network. Centralized localization also has difficulty with network topology changes due to node mobility or failure which also adds to the already high communications overhead. For many resource-constrained AWSN applications, a distributed approach is more suitable, as it is more efficient and scales better than a centralized approach. Furthermore, a distributed approach typically provides additional fault-tolerance in the presence of node or communication failures [SPS02].

The hardware and algorithms used to perform localization are influenced by the cost, size, and power constraints of wireless sensor platforms used in AWSNs. Crossbow Technology produces wireless sensor platforms, commonly referred to as motes, which have a form factor of 25mm and are sized to fit on a 3V coin cell battery [Ric05]. The smart dust chip or "Spec" mote, developed by researchers at UC Berkeley, is a single wireless chip with integrated CPU, memory, and RF transceiver with a form factor of approximately 5mm [Bra05]. The current cost of these motes is about $100 each [Ric05], but the number needed for many applications dictate that they must ultimately be a fraction of that price; making the cost of a several

thousand node AWSN acceptable. Limited power is also a major concern in an AWSN application. Crossbow's MICA motes are designed to be energy-efficient, and draw about 8 milliamps per hour when processing data. Additionally, the RF transceiver on the MICA motes consumes less than 1 microamp when off, 8 milliamps when receiving and 25 milliamps when transmitting [Ric05]. Given a battery rated at 1,000 milliamp-hours, a mote can operate with constant processing for 30 to 125 hours, depending on the time spent transmitting and receiving. Most implementations however, use a power-aware MAC, routing, and sensing protocols so the CPU and receiver are in low power mode or sleep mode much of the time; therefore, there is potential for these sensor platforms to operate for a year or more. By implementing energy-efficient protocols and algorithms that incorporate small, inexpensive, and energy-efficient hardware, a localization algorithm that fits within the cost, size and power constraints of an AWSN can be achieved. The remainder of this chapter considers the size and cost issues mentioned above, along with the following three critical metrics when discussing different methods of localization:

(i) Position accuracy or error,

(ii) Total time required to achieve desired position accuracy, and

(iii) Total network energy required to achieve localization.

Localization methods generally follow a four-step process: (1) Determine Range: determine node-to-node distances or proximity; (2) Disseminate Ranging Data: share predetermined attributes of ranging data between nodes; (3) Estimate Node Positions: perform position estimation to derive positions of *unknown* nodes; (4) Refine Node Positions: iteratively refine position estimations. The next four sections describe these steps in greater detail, discussing several techniques and the challenges associated with each.

2.3 Ranging

Ranging is the process of acquiring distance or proximity measurements between two nodes. It is the first and most important step in range-based localization, since inaccurate ranging measurements will ultimately result in poor position estimates. Ideally, ranging hardware and methods are highly-accurate, long-range, energy-efficient, low-cost, and small to achieve precise and efficient localization performance.

Received Signal Strength, Time of Flight (TOF), and Angle of Arrival (AOA) are physical characteristics of a signal often used to obtain distance and angle measurements in a wireless sensor networks. Many ranging algorithms using these signal characteristics have achieved accurate distance measurements but either require expensive hardware, lack long-range capabilities, or as with received signal strength, results are based on idealistic assumptions about signal propagation characteristics [HE04]. The next three sections discuss variations of the three ranging methods as well as their strengths and weaknesses

2.3.1 Received Signal Strength. Received Signal Strength Indicator (RSSI) measurements derive pair-wise node distance measurements using RF signal attenuation in a mathematical propagation model to derive distance. RSSI is an attractive option as it is relatively inexpensive, simple to implement, and typically does not require additional hardware to implement. Many RF transmitters/receivers have RSSI capabilities built in; eliminating additional hardware and node complexity while allowing the localization algorithm to benefit from normal network traffic. However, RSSI measurements are susceptible to several sources of interference such as fading, multi-path, and non-line-of-sight reception, resulting in inaccurate distance measurements with errors increasing with node separation distance. The biggest sources of error in an outdoor environment measurement errors due to obstacles, reflections, and variations in altitude [SHS01]. For example, the usable transmission range of an RSSI signal can range from 10m at ground-level to around 100m at a height of 1.5m [SHS01]. Simple implementations assume isotropic spherical radio propagation, when

in actuality this is not the case. However, since radio signal strength samples are normally distributed, errors due to non-isotropic radio propagation can be overcome by taking the average of several (30 or more) samples [Whi02]. Although this solves the problem of measurement errors due to non-isotropic radio propagation, it does not eliminate errors due to systemic effects. Additionally, averaging samples results in a significant increase in the number of ranging transmission required to localize. The transmit power of the source is an issue since it can often decrease as the power available at the source decreases with time. Wireless sensor networks, however, also possess two properties that help overcome RSSI range measurements errors. They are [SRB01]

(i) A dense interconnectivity of sensor platforms which leads to redundant range measurements, and

(ii) Long observation times in static wireless sensor networks can remove fast-fading effects through integration.

Off course, not all AWSNs will have these properties. In these cases, a probability distribution function of the distance corresponding to the RSSI of beacon packets may result in more accurate distance estimates [SR04]. Ultimately, RSSI is a viable ranging option in outdoor wireless sensor network applications, with near ideal or isotropic environmental conditions, where fine grained position accuracy is not required.

2.3.2 Time of Flight (TOF). Ranging methods based on TOF are the methods of choice for networks requiring fine-grained position estimate accuracy. Various methods use RF, audible, ultrasonic, or a combination of signals to perform TOF measurements. The method of measuring TOF using one signal is referred to as Time of Arrival (TOA), while methods that calculate the time difference between arrivals of multiple signals is referred to as Time Difference of Arrival (TDOA). Since the speed of a radio wave is equal to the speed of light (in a vacuum) and is constant over short distances, the time between transmission and reception of a signal can be

used to calculate the distance the signal traveled. This is the method used in GPS. This approach requires the transmitter and receivers clocks be closely synchronized to correctly measure TOF. Otherwise, even the slightest time difference between a pair of clocks can result in extreme errors in range estimates. In fact, if the receiving node's clock is ahead of the sending, the calculation may actually result in a negative time of flight.

A TOA approach that avoids clock synchronization altogether is TDOA using both radio and audible signals. Since the propagation speed of a radio signal is approximately 106 times faster than the speed of sound, the difference of the arrival times of the sound and radio signals is good estimate of the time of flight. As long as the two signals are transmitted simultaneously, a receiver can calculate the TOF differences without synchronization with the transmitter. Unfortunately, channel noise, echoes, and obstructions often make TOF measurements inaccurate and unreliable. For example, accurately detecting the beginning of an audible signal is problematic, and the ability to generate a sound signal with a sharp rising envelope requires certain hardware [SBM+04]. Furthermore, accurate detection of a noisy signal is difficult. However, over-sampling of acoustic signal and signal processing techniques such as filters and heuristics can increase the signal-to-noise ratio (SNR) [SBM+04]. An increase in SNR can result in less than 10 cm average ranging error for ranges up to 10m [SBM+04]. Although this is less sensitive to background noise and has a longer range than previous implementations, it comes at the expense of increased computational, communications, and associated power costs.

To overcome the inherent problem of measuring the TOF of an audible signal, experimental systems such as AHLoS [SHS01] and DOLPHIN [MSK04] implement TDOA with radio and ultrasound signals and achieve accuracies within 2 centimeters for node separations under 3 meters. Similar to acoustic ranging, errors in ultrasonic ranging are minimized through multiple-sampling, filtering and applying heuristics. However, ultrasonic transmitters/receivers are highly directional, typically only having a 120^o field of transmission with signal degradation towards the outer edge of the

field. This results in a single ultrasound receiver only being able to detect signals accurately in a relatively small field of view ($\sim 90^o$). Because sensor platforms in a AWSN perform range in every direction, multiple ultrasound transmitters/receivers must be used. AHLoS uses 4 pairs, enabling it to perform "accurate" ranging in almost every direction. Due to the resulting increase in the overall form factor, power consumption, and cost of the sensor platform, TDOA using ultrasound will not be a viable solution for some AWSN applications.

Although ultrasound ranging achieves centimeter accuracy for short distances, it also can have large errors or fail completely in low node density and non line-of-sight conditions [SHS01]. Additionally, to achieve long ranging capabilities (10+ meters), a more powerful ultrasonic ranging module would be required, resulting in a higher cost and added power consumption. Consequently, this approach is best suited to high-precision localization for infrastructure based networks as the added size, cost, complexity, and power requirements make it unsuitable for many AWSN applications.

2.3.3 Angle of Arrival (AOA). AOA exploits the ability of nodes to sense the direction from which a signal is received to aide in position estimation and to provide node orientation. An antenna array or several ultrasound receivers detect the angle at which a signal is received relative to its own axis of reference [NN03b]. The Cricket Compass Project [PCB00] obtains the arrival angle via range estimates (using TDOA with ultrasound and RF) at two local ultrasound receivers separated by a known distance. Figure 2.2 illustrates how to determine a node orientation with respect to a transmitting node given the two measured ranges, $x1$ and $x2$, the distance between the two local receivers, L, and the distance, x [PCB00]. The distance x can be calculated exactly using the laws of Cosines and Sines or simply estimated as the average of distances $x1$ and $x2$. Given x, θ can be determined using

$$\theta \quad = \quad \arcsin\left(\frac{x_2^2 - x_1^2}{2Lx}\right). \tag{2.1}$$

The AOA technique does not eliminate the need for ranging. In fact, it requires precise ranging hardware such as TDOA using ultrasonic signals. Additionally, AOA does not provide any better error propagation performance than other techniques. A significant advantage of using AOA is that node orientation could be very helpful in applications where organized node movement is required. However, the resulting increase in size, cost, and power-consumption makes AOA an expensive and unattractive option for many resource constrained AWSN applications.

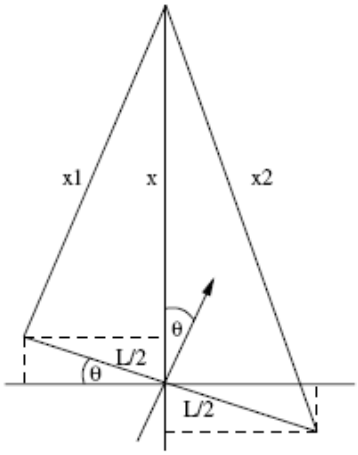

Figure 2.2: Two-Dimensional AOA [PCB00]

2.3.4 Range-free. Range-free localization methods derive proximity measurements from non-distance information. Thus, the "range-free" designation. These measurements may include mere radio communications connectivity [SPS02] or a comparison of received signal strength [LWH04].

The level of localization precision or granularity is always application-dependent. Therefore, the added hardware cost for range-based ranging solutions may not always be appropriate, since many range-free solutions can sometimes achieve acceptable precision at a lower cost.

2.4 Dissemination of Ranging Data

Given ranging data along with the estimated position of known nodes, the next step in the localization process disseminates and merges data to determine accurate distance estimation from *unknowns* to *anchors* in the network.

Depending on the AWSN application and its purpose, the goal of localization is to derive absolute, relative, or virtual positions or coordinates. Many applications require absolute coordinates as the users require specific global locations of sensed events. However, in some AWSN applications precise location information is not necessary, and relative or virtual coordinates will therefore suffice. Relative coordinates are typically derived from range-based localization, while virtual coordinates, which are not as fine-grained, are typically derived from range-free localization algorithms. Some localization algorithms, such as the Map-growing algorithm [LSS04], initially produce a network topology with only relative positions for *unknowns*, and then transform the relative positions into a global position using the global coordinates of 3 or more *anchor* nodes.

Regardless of the type of coordinates required, the network must successfully merge and propagate position, range, directionality, and/or connectivity data. The amount of data shared is mostly dependent on the complexity of localization algorithm. Some of the simpler localization algorithms require each node to obtain ranging information from 1-hop neighboring nodes, while the more complex algorithms require nodes to have almost total knowledge of the pairwise node distances for every node in the network. For this reason, the localization process can quickly become extremely costly in terms of communications and power required to disseminate the necessary data.

The position estimation techniques discussed later in Section 2.5 rely on a considerable amount of critical information that can typically only be obtained through a network-wide distribution and sharing of ranging and node position data. Additionally, it is crucial that multi-hop distances between *unknowns* and *anchors* be accu-

rately merged and represented through the sharing of this information. The biggest challenge of this process is to avoid cumulative errors that result in large position estimation inaccuracies.

The Ad-hoc Positioning System (APS) is a class of distributed localization algorithms that propagate distance information to *anchors* throughout the network so all *unknown* nodes can obtain distance or angle estimates to 3 or more *anchors* [KMS+04]. The DV-distance, DV-hop, Euclidean, and DV-coordinate algorithms are common APS algorithms. The main difference between the four algorithms is how the distance information is represented as it propagates throughout the network. DV-distance algorithms propagate pairwise node distances while maintaining the minimum path lengths to *anchors*. The distance to a particular *anchor* is determined by the summing the distances in the shortest path to that *anchor*. DV-hop on the other hand, propagates average distances per hop while maintaining the minimum hop count to *anchors*. Distance estimates to any particular *anchor* in the network are obtained by multiplying the hop count by the average distance per hop. The Euclidean algorithm concurrently computes distances between *unknowns* and *anchors*; maintaining only distances from *unknowns* to *anchors*. A comparison of these algorithms show that each perform well in different scenarios [LR03] [NN03b] [KMS+04]. For instance, DV-hop and DV distance algorithms usually require isotropic networks and uniform node density to achieve accurate position estimation, while Euclidean techniques are more resilient to irregular networks, but require a higher degree and *anchor* density to achieve accurate position estimation [KMS+04] [LR03].

DV-coordinate is an APS localization algorithm that has every node initially derive its own relative coordinates or map one-hop neighbors. Instead of passing distance estimates to *anchors* like the Euclidean algorithm, DV-coordinate passes local coordinate information from node to node. Receiving nodes transform the received coordinate information into their own coordinate system; thereby building the map. Localization is accomplished once a node establishes a local map with at least three non-collinear *anchors*. This is very similar to the Self-Positioning Algorithm

(SPA) [CHH01], which establishes relative coordinates for an entire network. Additionally, SPA adjusts and maintains the relative coordinates of a network with low to moderate node mobility. Two major weaknesses of these algorithms is the high communications cost and the requirement for full-network connectivity to *anchors* to derive position estimations. This ultimately leads to needing a much higher *anchor* density.

DV-Bearing and DV-Radial are angle-based localization algorithms similar to the above mention distance-based APS algorithms [NN03a]. They are most similar to the Euclidean algorithm as they propagate either true bearing angle estimates, or true bearing and radial angle estimates from *unknowns* to *anchors* in a network. The methods rely on triangulation to calculate intermediate angle estimates before propagating them throughout out the network.

It is crucial for these and all ranging propagation techniques and algorithms to be efficient and robust to limit unnecessary communications and ensure scalability to varying network sizes, topologies, and irregularities. Additionally, they must accurately propagate and represent distances from *unknowns* to *anchors* to minimize position estimation errors.

2.5 Position Estimation

This section discusses the fundamental concepts of position estimation techniques and how they are applied to localization in a AWSN.

2.5.1 Lateration. Trilateration estimates the 2-dimensional position of a node by measuring its distance from 3 or more reference points. Figure 2.3 illustrates the concept of trilateration, given the distances to and locations of 3 *anchors*, to determine the location of a single *unknown* node. Simply drawing circles with radii equal to the estimated distances to the respective *anchor* nodes results in a single intersection point where the *unknown* node must lie. Calculating the position at this point can be accomplished using linear algebra. One common method described

in [LR03] involves setting up a system of linear equations and solving it using a standard least squares approach shown in Appendix A.

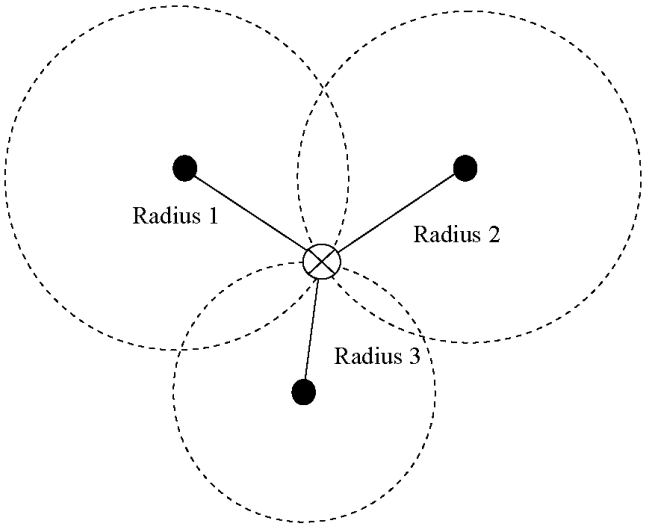

Figure 2.3: Obtaining 2D Position Estimation using Trilateration (adapted from [HB01a])

Given distance measurements to four or more non-collinear *anchors*, lateration can be used to obtain three-dimensional position estimation of *unknowns*. More than 4 ranging measurements to *anchors* can be used to further improve the accuracy of position estimation by averaging the lateration results of all combinations sets of 3 (2-dimensional) or 4 (3-dimensional) *anchor* nodes.

Trilateration requires the three referenced *anchor* nodes to be non-collinear, since collinear references would result in two possible points of intersection instead of one. Additionally, references that are near-collinear may be perceived as collinear due to errors in ranging measurements. Thus, many localization algorithms check to see if the three reference nodes form a triangle with a minimum angle greater than a certain threshold. For example, the Map-growing algorithm uses a minimum angle threshold of 30 degrees [LH05]. Given the distance between three nodes (a, b, and c), the Law of Cosines determines the three angles of the formed triangle

$$a^2 \;=\; b^2 + c^2 - 2bc\ \cos(A). \tag{2.2}$$

Similarly, some algorithms use a simpler non-collinear test that ensures the sum distance of the smallest sides of the triangle is greater than the third side multiplied by a given threshold [LR03].

Multilateration is the process of estimating a position based on the range measurements of three or more *anchors*. Atomic multilateration is the simplest form of multilateration (represented in Figure 2.4a), which only uses ranging measurements from *anchors* to estimate position of *unknowns*. A more robust form of multilateration referred to as iterative multilateration consists of using estimated positions of *unknowns* to serve as *anchors*; allowing achievable localization in light of low *anchor* densities. Iterative multilateration is illustrated in Figure 2.4b where node U2 may use the two neighboring *anchor* nodes plus the estimated position of node U1. Figure 2.4c illustrates collaborative multilateration, where nodes U1 and U2 are able to collaborate and share neighboring *anchor* information to estimate both of their positions.

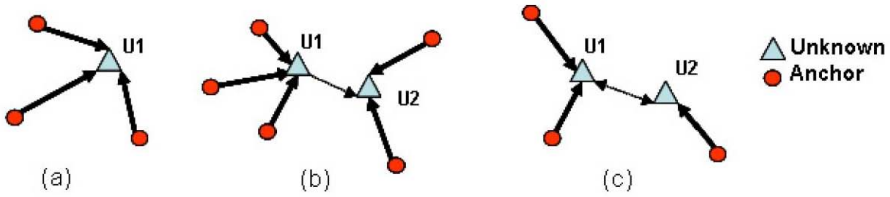

Figure 2.4: Atomic, Iterative, and Collaborative Multilateration (Adopted from [SHS01])

Both iterative and collaborative multilateration are more robust than atomic, but are prone to more significant error accumulation, due to the use of localized *unknowns* or the sharing of information between neighboring *unknowns* for position estimation of other *unknowns*.

2.5.2 Multi-Dimensional Scaling (MDS). MDS is any procedure that uses a set of distance measurements between several points to find a set of relative coordinate values [Lee05]. MDS is a position estimation approach that derives relative position estimates and transforms them into global coordinates [LSS04]. In classical MDS, a

distance matrix consisting of an all-pairs-shortest-path data to one or more nodes is required.

A more robust form of MDS, iterative MDS (otherwise known as Least Squares Scaling (LSS)) is more robust than classical MDS since it always generates relatively accurate position estimates even with inadequate and inaccurate distance information [JZ03] [JZ04]. Unlike classical MDS, LSS is capable of estimating positions even with missing distances in a given set of all-pairs-shortest-path distances. The greater the number all-pairs-shortest-path distances, the more accurate the position estimation results are. If just one or two distances are used, large position errors will occur [SPS02]. Additionally, LSS accepts the assignment of different weights for distance measurements depending on their confidence levels as well as constraints on node separation. This ultimately improves the accuracy of the resulting position estimation as bad points are discarded and greater weight is given to points with higher confidence [KMS+04].

Most forms of MDS are implemented using a centralized algorithm. This is mainly due to better position accuracy is achieved by having a larger amount of ranging and position data. However, distributed approaches using LSS can also achieve very good accuracy [KMS+04] [SPS02]. For example, a distributed algorithm, N-hop Multilateration, sets up a global non-linear optimization problem and solves it using iterative least squares scaling [SPS02]. N-hop Multilateration essentially performs LSS through a series of collaborative and iterative multilateration steps performed in a repetitive sequence. The algorithm provides a method for establishing a random refinement sequence that can be repeated in order to ensure all nodes conform to the global gradient. Otherwise, neighboring nodes will likely arrive at a local minimum with erroneous position estimates due to "local oscillation" [SPS02].

2.5.3 Min-Max. An alternative to Lateration and MDS is the Min-Max position estimation approach discussed in [SPS02]. This technique requires less computational complexity than the previous approaches but results in a less precise posi-

tion approximation. Figure 2.5 illustrates the Min-Max technique where the *unknown* node is bounded in the X and Y coordinates by the distance measured to each of the three surrounding *anchors*. Once the minimum and maximum X and Y coordinates are determined, the node estimates its position to be in the center of these minimum and maximum X and Y values respectively. One study concludes that Min-Max can outperform Lateration in experiments with large ranging errors (greater than 10% standard deviation) [LR03].

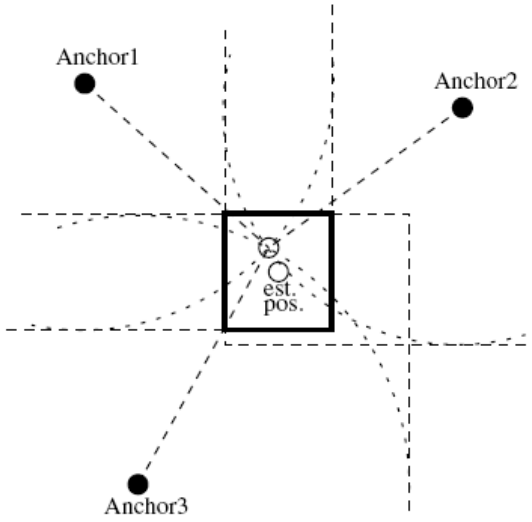

Figure 2.5: Min-Max Position Estimation [SPS02] [LR03]

2.5.4 Angle of Arrival (AOA). Triangulation is the most common form of angulation used to perform two-dimensional position estimation of an *unknown* node given angle of arrival (bearing) estimates to two or more *anchors*. There are several methods for performing position estimation using triangulation. The chosen method is mostly driven by the hardware capabilities of the sensors in the network.

For example, given all nodes are capable of maintaining and sensing received angles with respect to one global, constant, reference vector or axis (i.e, magnetic north) [HB01b], then triangulating the position of an *unknown* only requires knowledge of the angles of arrival for signals from two *anchor* nodes. This solution requires

the *unknown* to first use the point-slope methods to determine the equation of the lines between it and the two *anchor*. Then the position of the *unknown* is given by the intersection of the two lines, which can be computed using linear algebra.

However, if the sensor platforms not equipped with compass-like hardware, the process is more complicated. The most straightforward approach uses the distance between pairs of *anchors* since it can easily be calculated using the Pythagorean Distance Formula as well as the fact that *anchors* are capable of determining the angles formed in their respective corner of the triangles. Figure 2.6 illustrates this approach. Note that a single common reference vector (node B) is not required for this approach to work. Given that an *unknown* knows the distance between two neighboring *anchors*, side AB, as well as the bearing to each, θ, and the angles α and β, the Law of Sines is applied to determine the distances between the *unknown* and both *anchor* nodes. The same steps are applied to one or more subsequent neighboring *anchor* nodes, node C for example, until the requirements for performing trilateration are met. This technique ultimately transforms the triangulation problem into a simpler trilateration problem.

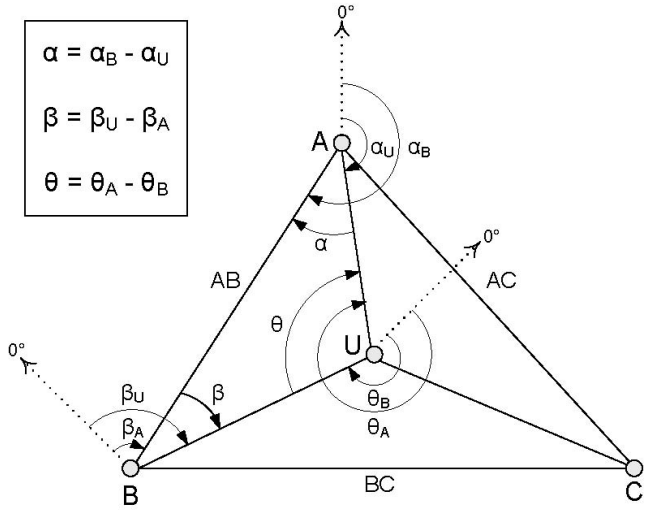

Figure 2.6: Obtaining 2D Position Estimation using Triangulation

2.5.5 Range-free. Range-free position estimation techniques rely on the results of trigonometry and/or geometry to derive position estimations from node connectivity information. Connectivity criteria can range from mere radio communications connectivity to a comparison of received signal strength [LWH04].

Figure 2.7 shows two common convex position estimation approaches using triangular and circular regions. It demonstrates how positions of *unknowns* nodes can be confined to convex regions using connectivity constraints to *anchors*. Specifically, the *unknowns* are constrained to the gray areas. Many connectivity based algorithms perform a center of gravity calculation to best approximate the position of the *unknown* node.

Comparing Figures 2.7a to 2.7b and Figures 2.7c to 2.7d, it is easy to see how a more precise position estimation can be derived given a greater number of *anchors* with a favorable geometry. The main limitation of many range-free localization approaches is that they require a high *anchor* density to achieve acceptable position accuracy. For many AWSN applications, this makes a range-free approach impractical.

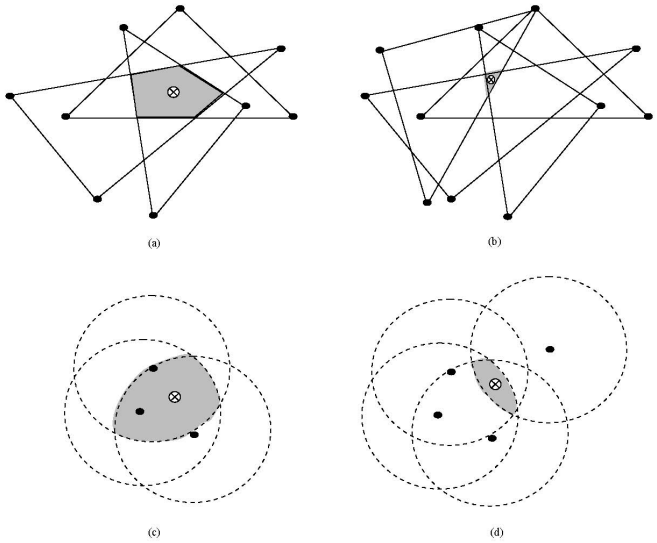

Figure 2.7: Obtaining 2D Convex Position Estimation using Connectivity

2.6 Position Refinement

After completing initial position estimation, every node in the AWSN has an estimated position in the overall network topology. The average position error is dependent on the accuracy of the ranging measurements as well as the position estimation method. However, initial position estimation errors can be significantly reduced by additional post processing of initial estimates [SRB01] [SPS02] [LR03].

In general, the accuracy of position estimates can be greatly improved through an iterative refinement process, where each node uses the most recent position estimates and ranging measurements of neighboring nodes to recompute their own position. The Hop-TERRAIN algorithm [SRB01] uses a refinement step that periodically recalculates position estimates based on updates from neighboring nodes. To avoid erroneous position estimates, it computes and uses confidence levels of position estimates as well as ensuring new position estimates are still within the reach of the original estimated distances to *anchors*. The combination of these techniques ultimately maintain a global gradient and prevent erroneous position estimates. Hop-TERRAIN reduced its original position error of 39 percent of the maximum range to just 5 percent after 25 iterations [SRB01]. Alternatively, a refinement method that maintains a global gradient and prevents "local oscillation" of neighboring nodes, also results in improved position estimates. N-hop Multilateration algorithm accomplishes this by using a repeatable refinement sequence [SPS02].

Data correlation from sensed events can further refine position estimates. Specifically, Online Localization [GKLP04] and Manifold Learning Localization [PH04], opportunistically use information gained through the normal mission of a AWSN. Hence, their strength is the use of data inherent in AWSN applications which ultimately improves localization accuracy without additional communications costs.

2.7 Relevant Research

Node Localization in wireless sensor networks has been an area of active research in recent years, with many new and innovative approaches focused on achieving efficient, accurate, and cost-effective localization in ad-hoc networks. Although much of the research has focused on experimenting with "ideal" isotropic networks, some have used more irregular or anisotropic AWSN applications. Some common sources of irregularity in a network are initial node placement, obstructions and irregular network shape, and the addition of mobile nodes. This section presents current research in these different dynamics and challenges of AWSN localization. Lastly, the two algorithm used for this research, APS-Euclidean [NN01] and Map-Growing [LSS04], are introduced and discussed in terms of their performance in anisotropic and mobile networks.

2.7.1 Anchor Placement. *Anchor* assisted localization algorithms rely on network node connectivity and *anchor* placement [SHS01]. The probability that a node in the network has a connected degree of 3 or more is a key characteristic to consider when planning AWSN deployment. The probability a single node will have a degree greater than a desired number, d, for large values of N is

$$P(d) = \frac{(NP_R)^d}{d!} e^{-(NP_R)} \quad and, \tag{2.3}$$

$$P(d \geq n) = 1 - \sum_{i=0}^{n-1} P(i), \tag{2.4}$$

where N is the number of nodes in a network, $P_R = \frac{\pi R^2}{L^2}$ is the probability a node is in transmission range of another node, R is the transmission range in meters, and L is the approximate side length (in meters) of a square network [SHS01]. The percentage of *anchors* needed to meet the minimal requirement of atomic multilateration decreases as the average node degree increases.

Some AWSNs use self-configuring *anchor* networks [BHE00] [BHE01] [BHET04]. This algorithm requires a given *anchor* density to achieve a quality estimate of local-

ization [BHET04]. Therefore, the probability a packet is successfully received without interference in a self-configuring network is

$$P_{success} = \frac{T_x}{T}\left(1 - \frac{T_x}{T}\right)^{\rho\pi R^2} \quad (2.5)$$

$$P_{collision} = 1 - P_{success} \quad (2.6)$$

where ρ is the anchor density, T_x is the transmission time, and T is the beaconing interval [BHET04]. Thus, the probability of a successful transmission decreases exponentially with increased anchor density. This results in an inability to efficiently localize in a timely manner. This leads to an anchor density threshold where the benefits of the increased number of *anchors* is outweighed by the network congestion it causes.

Unfortunately, guaranteeing a deployed network has the desired isotropic *anchor* density is difficult and often not feasible. Thus, algorithms have been developed for networks with *anchor* densities that would be considered either too low or too high. For networks with low and medium *anchor* densities, the HEAP algorithm detects regions with poor localization and selects optimal areas for placing additional *anchors* [BHE01]. This algorithm has all the *anchors* in a network share information with each other (directly or indirectly), followed by each individual *anchor* using that information to determine candidate points for new *anchor* placement. Every *anchor* forwards their candidate points to a single central node, referred to as the placer. The placer is responsible for choosing the best candidate points for new anchors; ultimately improving the distribution of anchors in the network. A major drawback of this algorithm is that it relies on a single placer node; creating a single point of failure. Additionally, HEAP assumes new nodes can be placed in the chosen locations. Although not mentioned in the HEAP research, a similar or modified algorithm could take advantage of mobile nodes to move already deployed *anchors* to new positions to improve the topology of the deployed *anchors*.

Since adding new anchors to a deployed AWSN is not always feasible, AWSNs can be deployed with a higher anchor density than desired. In these instances, the STROBE algorithm (Selectively TuRning Off BEacons) can "reduce channel contention while exploiting the spatial diversity and redundancy of densely deployed" *anchors* [BHE01]. STROBE nodes have three states: Designated, Voting, and Sleep. In the "Designated" state a node is an active *anchor* and sends periodic beacons, while in the sleep state, a node is inactive for the set period of time. Upon entering the voting state, a node is an active *anchor* for the designated period of time. At the end of the designated time period, a node determines if it should transition to the "Designated" state or the "Sleep" state based on the number of active anchors in its neighborhood and the anchor density threshold. If the number of active anchors is less than the *anchor* density threshold, the node automatically transitions to the "Designated" state. However, if the number is equal to or greater than the anchor density threshold, the probability it transitions to the "Designated" state is

$$p = \frac{\mu_{thresh} - 1}{\zeta} \tag{2.7}$$

where μ_{thresh} is the anchor density threshold and ζ is the number of active anchors in the 1-hop neighborhood. The probability a node transitions to the sleep state is therefore (1 - p) [BHET04]. Techniques such as STROBE maintain uniform anchor density, while minimizing and fairly distributing power consumption at each beacon.

2.7.2 Irregular Network Shape. One common assumption many localization algorithms make is network topology is isotropic, i.e., the properties are identical in all directions. Some sources of anisotropic characteristics are irregular or non-uniform network topology, irregular geography, and obstructions such as buildings, obstacles, and foliage. These types of network irregularities strongly influence localization performance. Therefore, a robust range-based localization algorithm must accurately estimate geographic distances from *unknowns* to *anchors* in anisotropic sensor networks to achieve accurate position estimates.

Many localization algorithms such as APS rely on high node and/or *anchor* densities to localize in an anisotropic network. The performance of some of the APS algorithms to a Map-growing algorithm in different anisotropic network configurations has been studied [LSS04]. The network configurations varied the initial node distribution (Normal Grid with variance & Random Normal), as well as the network shape (Full-Square, O-Shaped, and C-Shaped). The variation in node distribution models the randomness of an AWSN deployment, while the shape-variation models topological irregularities due to geography, man-made structures, or node failures. The localization error of the DV-distance algorithm ranges from under 30 of maximum range in the Normal Grid configuration shown in Figure 2.8a, to over 220 percent in a C-shaped random network configuration as shown in Figure 2.8b. Additionally, given an average node degree of 12.3, the Euclidean algorithm requires a high anchor density (30%+) in all network configurations (isotropic or anisotropic) to localize at least 80% of the deployed nodes.

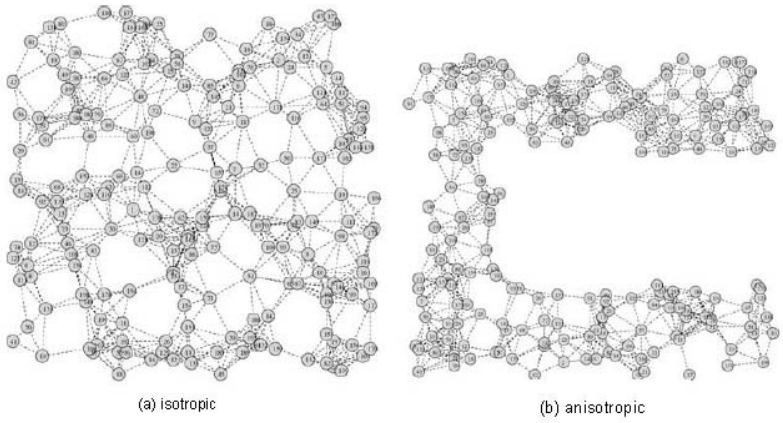

Figure 2.8: Isotropic versus Anisotropic Network Configurations [NN03b]

2.7.3 Node Mobility. Many *anchor* based localization algorithms do not model or plan for node mobility [SPS02] [NN03b] [BHET04]. However, most can achieve and maintain localization in mobile networks by simply refreshing location estimates frequently when a node moves. Unfortunately, these frequent updates in-

crease the communication overhead and corresponding power costs of performing localization.

Dynamically localizing mobile nodes with stationary *anchors* make achieving localization difficult, since as node speed increases so do position estimation errors [BM02]. However, Monte Carlo Localization (MCL), a technique originally developed for robotics localization, takes advantage of node mobility to improve position estimation accuracy while reducing the required number of *anchor* nodes [DFBT99]. MCL incorporates a position prediction and an update phase, which occurs between each node step or movement. The prediction phase derives a new position estimate based on adding the uncertainty due to the step/movement to the previous sample of possible positions. Prediction is followed by an update phase where new measurements or observations, such as the presence or absence of landmarks, are considered to filter and update the set of new potential positions. Repeating this process allows a node to continually update and even improve its position estimate. This method also improves localization accuracy in networks with anisotropic radio propagation. Specifically, in networks with variation in the maximum radio range, an implementation of MCL adapted for use in AWSN node localization, achieves 35% to 65% lower position estimate error compared to other proposed range-free algorithms [HE04].

Alternatively, some network applications may not require mobile nodes to know its current position. In these cases, a mobile node may simply send a *hello* message upon stopping, and after receiving replies from its new neighbors, it estimates its position as it normally would. This avoids unnecessary communications overhead and a potential increase in position estimation errors due to re-estimating position while mobile.

Anchor nodes are not always stationary [SHS04] [PBDT05] [SR04]. For example a single mobile *anchor* can assist *unknown* nodes to localize [SR04]. Furthermore, the mobile *anchor* does not have to be a sensor platform; it could be a person, animal, or even an aircraft. If these approaches could work, the communications cost to

achieve localization would be drastically reduced as the need to propagate ranging measurements is eliminated and the number of beacon signals is reduced to one. One major challenge is the *anchor* node must follow a trajectory that covers the entire deployment area while ensuring each point receives at least three non-collinear beacon signals [SR04].

2.7.4 Algorithms. Although many algorithms are not designed for node mobility or network irregularity, several are still capable of localizing in such scenarios; though often with degraded performance. Map-growing and APS-Euclidean algorithms can be so modified. The remainder of this section discusses these algorithms along with how network irregularity impacts them.

2.7.4.1 APS-Euclidean. Of all the APS algorithms, Euclidean is said to be more accurate and more predictable in anisotropic networks [NN01]. This version of the APS algorithm propagates the true Euclidean distances to *anchors* while the other algorithms use hop counts as distance estimates to anchors. The APS-Euclidean algorithm has every node concurrently estimates distances to *anchors* in the network [NN01] [NN03b]. Nodes must communicate with immediate neighbors; sharing all estimated distances to *anchors*, distances to one-hop neighbors, and anchor distance estimates of one-hop neighbors. Thus, the Euclidean algorithm uses second-hop information. After obtaining distance estimates to three or more non-collinear anchors, a node can estimate its position. However, upon receiving information regarding unknown *anchors*, a localized node continues to estimate its distance to them. After estimating the distance to new *anchors*, a localized node propagates the estimates to its neighbors, to further assist network localization.

Although APS-Euclidean is said to perform well in anisotropic networks of varying shapes [NN03b], in networks with low connectivity, only a small percentage of nodes can localize due to the low neighbor degree and the resulting inability to estimate distances to *anchors* [LR03]. APS-Euclidean incorporates node mobility be-

tween a newly positioned mobile node and its new neighbors. A mobile node sending a *hello* packet in its new position, will result in responses from its new neighbors.

2.7.4.2 Map-Growing. The Map-growing algorithm is an incremental localization algorithm [LSS04]. This algorithm is more complex than APS-Euclidean in that it has three phases. The first is the election phase, where a single starting node is elected and two subsequent neighboring nodes are chosen to make up a center island or triangle of relative coordinates, with the starting node having the relative position of (0,0). In the second phase nodes incrementally estimate their relative positions as the relative map grows. Lastly, the third phase floods the global and relative positions of every *anchor* node in the network so every node can perform a transformation from the previously estimated relative coordinates to global coordinates. Like the APS-Euclidean algorithm, Map-Growing requires one-hop and two-hop neighbor information to satisfy the position estimation requirements.

Unlike APS-Euclidean, given a connected undirected network, Map-Growing reaches 100% global localization with as few as three non-collinear *anchor* nodes for both isotropic and anisotropic networks [LSS04]. Additionally, just like APS-Euclidean, the impact of a mobile node is also constrained to a mobile node re-entering the network and its new neighbors. However, during the election phase, it is not desirable to have mobile nodes as it would require a more complex election algorithm.

Regardless of the algorithm, node mobility increases communications overhead, and can possibly impact network connectivity and shape. Depending on the type of mobility (intelligent, organized, or purely random), this effect may or may not be controllable. In the simplest case with purely random mobility, nodes continue to move until successfully localized. The resulting network topology will most likely be more highly connected than originally deployed.

2.8 Summary

Node localization is a fundamental problem in AWSNs. This chapter provides an overview of node localization in AWSNs. Additionally, some of the characteristics, constraints, and difficulties associated with achieving accurate and efficient node localization are discussed. The chapter presented in detail the four general steps of localization, ranging, ranging dissemination, position estimation, and position refinement. Lastly, this chapter discussed current research and challenges dealing with achieving localization in non-isotropic and mobile AWSNs.

III. Methodology

3.1 Introduction

This chapter defines a methodology to evaluate the effects of network shape, network degree, and node mobility on AWSN localization. The overall experimental design is discussed in detail, and the information needed to duplicate the experiment is presented. Section 3.2 defines the problem and discusses the goals, hypothesis, and approach used. Section 3.3 describes the system under test. Section 3.4 defines the system services, and Section 3.5 describes the workload presented to the system. Section 3.6 explains the metrics which are observed in the experiments. Sections 3.7 and 3.8 explain the parameters and factors. Section 3.9 explains the evaluation technique and implementation details of the Map-Growing and APS-Euclidean localization algorithms. Section 3.10 describes the experimental design, and Section 3.11 is a summary of the chapter.

3.2 Problem Definition

Localizing nodes in an AWSN is more difficult when conditions such as network shape irregularities (due to geographic/environmental conditions), varying network topologies (node degree and node distributions), and mobile sensor platforms are introduced into the system.

3.2.1 Goals & Hypothesis. Operating for extended periods with limited battery power is a major concern and challenge for AWSN applications. The main goal of this research is to determine the effect of network shape, node degree, and node mobility on the data communications costs and associated power required to localize nodes. The hypothesis of this research is that the more irregular or dynamic the network configuration, the more communications and power required to achieve localization.

Another objective of this research is to examine the performance differences between an incremental algorithm and a concurrent algorithm in anisotropic and mobile

networks. To satisfy this objective, the incremental Map-Growing algorithm and the concurrent APS-Euclidean algorithm serve as representative algorithms for the experiments. Map-growing localization algorithms claim to achieve better coverage than APS-Euclidean in networks with low anchor ratios [LSS04]. However, according to the results in [NN03b], APS-Euclidean achieves better accuracy and comparable coverage for similar experiments. Besides examining the differences in coverage and position error, the communications cost differences of the two algorithms are determined. Since Map-Growing is a multi-phase incremental algorithm, it is hypothesized it will have higher communications overhead than the concurrent APS-Euclidean algorithm.

3.2.2 Approach. To achieve the above research goals, simulated AWSNs are observed under various operating conditions. Specifically, the bits transmitted and bits received responses are used to determine the associated power costs of performing localization.

The effects of the different levels of each factor on the mean response are computed and contrasted to determine if one level is significantly less or more than another. This will determine the effect of varying levels of network "irregularity" on the communications and respective power costs of each localization algorithm. Additionally, an analysis of variance will determine the percentage of variation explained by each factor and their interactions. Furthermore, the effects of the algorithm on the mean response are computed and contrasted to determine if the two algorithms perform significantly different.

Additionally, an analytical power model for both algorithms is developed, which provides a rough-order-of-magnitude estimate for the power costs associated with data communications and processing during localization. A multiple linear regression is performed on the Bits Transmitted and Bits Received responses to construct the power model. Using the regression equations a prediction of the number of the number of data bits transmitted and received can be obtained. Finally, a processing time is

estimated based on the number of instructions executed and included in the power model.

3.3 System Boundaries

Figure 3.1 shows the System under Test (SUT) and the Component under Test (CUT). The SUT is the Wireless Sensor Network, which consists of *anchors*, *unknowns*, and ranging error. The hardware components of the sensor platforms are considered outside of the SUT. The workload of the system include network shape, network topology, sensor mobility, and data packets. Radio communication is assumed to be isotropic and channel noise and interference are not modeled. Packet collisions due to network congestion can occur, however due to random delays being added to message transmission start times, they occur with very low probability. To simplify the model, retransmissions of collided messages are not modeled. This simplification is acceptable due to the redundancy in the algorithms. That is, the same data typically gets transmitted multiple times throughout the course of localization.

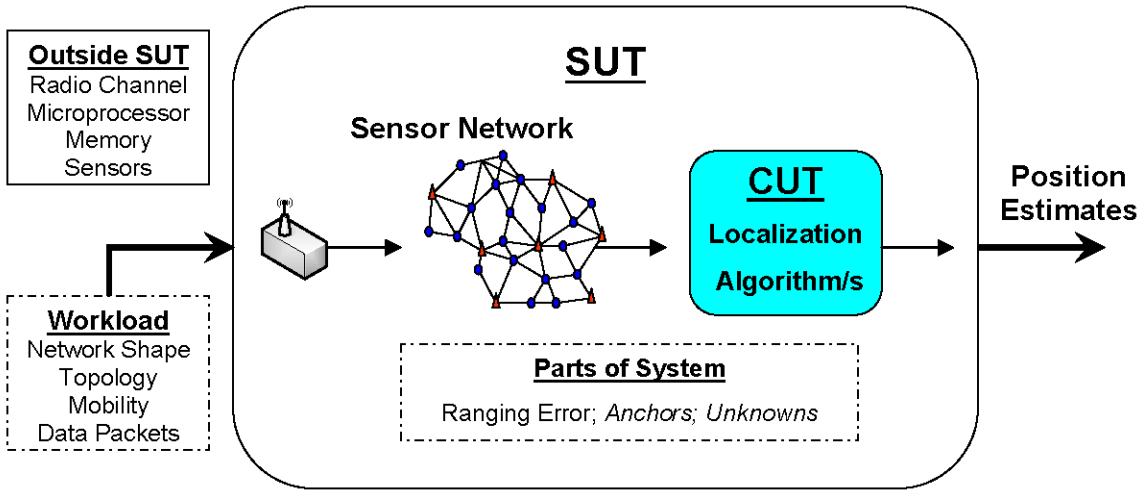

Figure 3.1: Graphical Representation of SUT and CUT

The component under test is the localization algorithms. Input to the algorithms include the location of each transmitting node in two-dimensional space, pair-wise distances between nodes within their maximum transmission range, and an applied

Gaussian error. Using this data, the algorithms estimate node positions when the conditions for localization are met.

3.4 System Services

The system provides a position estimation service for static and mobile nodes. A successful outcome is defined as a node deriving a position estimate within a desired accuracy. Failure is defined as the estimated position being outside a desired accuracy or a node being unable to estimate its position.

3.5 Workload

The workload for the system is the data that passes through the AWSN. This data includes ranging data and control data. Ranging data are signals broadcasted to neighboring nodes for the purpose of estimating distance between nodes. Nodes use control data to disseminate ranging data and other information to meet the requirements of the localization algorithm. Network shape irregularities (due to geographic or environmental conditions), varying network topologies (node degree and network distributions), and mobility of sensor platforms, all influence the amount of data that passes through the system and thus the workload of the system

3.6 Performance Metrics

The following metrics measure the performance of the localization algorithm:

- **Power Consumption**: Three critical power consuming functions of a sensor platform include transmitting, receiving, and processing. Therefore, the average number of bits transmitted, the average number of bits received, and processing time are the power consumption metrics used in this research to measure power consumed by the localization process. Bits Transmitted and Received responses are the sum of all data bits transmitted and received respectively for the entire network.

- **Accuracy**: Commonly, position accuracy is the most important performance metric of a localization algorithm. This study uses position error as a percentage of maximum ranging distance to measure the average accuracy of the localization algorithm.
- **Percentage of Nodes Localized**: The Percentage of Nodes Localized is defined as the percentage of nodes which have successfully localized. Random network configurations often result in disconnected or partially connected nodes. Additionally range error and the resulting propagation of range and position error sometimes results in nodes that do not satisfy position estimation requirements. Thus a localization algorithm will not always localize 100% of nodes in the network. Therefore, the length of the experiments are determined by the time it takes to reach 98% localization. However, if 98% localization is not achieved by a certain point, then the experiment ends regardless.

3.7 Parameters

The following parameters affect the performance of the system under test.

3.7.1 System.

- **Localization Algorithm**: The method for achieving localization has a direct impact on system metrics.
- **Ranging Method**: Localization algorithms use several different ranging methods. The particular method directly impacts maximum range capability and ranging errors, and thus the performance metrics. While not specified, the ranging method is assumed to have a maximum range of 3 distance units. The measurement error is modeled as a zero mean Gaussian random variable with a standard deviation of 10% of the actual distance between a pair of nodes. Antennas are assumed to be omni-directional.

- **Communications Channel**: Control messages and ranging data are sent through the network over a single radio channel. The maximum transmission range of the sensor nodes is 3 distance units.

3.7.2 Workload. Levels and variations in network shape, node degree, and node mobility constitute the workload of the system. Networks have 200 nodes distributed according to a uniform distribution. All experiments use a standard anchor ratio of 1 *anchor* for every 9 *unknowns* (i.e., 10%). *Anchors* are chosen at random from the 200 nodes and therefore are also randomly distributed throughout the network.

- **Network Degree**: Degree is defined as the average number of neighboring nodes within the maximum range of a node. In other words, it is the average number of one-hop neighbors of all nodes in the network. Degree directly affects the performance of a localization algorithm. In a network where all nodes, *anchors* and *unknowns*, are randomly distributed, degree and the ratio of *anchors* to *unknowns* ultimately determines the average *anchor* degree for any particular node in the network. If the *anchor* ratio is constant while degree increases, so does the average *anchor* degree.
- **Network Shape**: The overall area of a network and the physical layout defines the network shape. The geographic topology of the environment as well as obstacles or interference encountered determines the physical layout of the network. Irregularities in network shape can affect the accuracy of a localization algorithm as well as its ability to fully converge.
- **Node Mobility**: The addition of node mobility to an AWSN deployment directly impacts the performance of a localization algorithm. Mobile beacons can improve the coverage, accuracy, and power consumption of a localization algorithm [SR04] [PBDT05]. However, the addition of mobile *unknowns* will increase the time, communications, and power required to localize.

3.8 Factors

The four factors varied include the localization algorithm, network shape, average node degree, and node mobility. Table 3.2 shows the levels of each factor and the remainder of the section describes the levels in detail.

Table 3.1: Factors & Levels

	Factors	Levels
System Characteristics	Localization Algorithm	Euclidean, Map-Growing
Workload Characteristics	Network Shape	Full-square, C-Shaped
	Degree	Low, Medium, High
	Node Mobility	None, Low, Medium, High

There are two baseline experiments, one for each localization algorithm. The baseline experiments uses a full-square area consisting of 200 randomly distributed static nodes with medium average node degree.

- **Localization Algorithm**: Discussed in Section 3.1.1, this research examines the difference in data communications and associated power cost requirements of the Map-Growing and the APS-Euclidean algorithms.
- **Network Shape**:

  Full Square: Nodes are distributed throughout the entire square area. This factor level models an ideal deployed environment with no obstacles.

  C-Shaped: Nodes are distributed throughout the upper, left, and bottom of the square area. To maintain the same average node degree, the sides of the square area are scaled to ensure the nodes are occupying approximately the

same area as the *Full Square* networks. This factor level models an irregular network shape or an obstacle in the deployed environment.

- **Degree**: The average node degree has a direct impact on localization. High network degree intuitively results in localization that is more accurate. However, that same number of nodes covers a smaller area. The degree is varied to determine the effect it has on power consumed during the localization process. Degree is measured as the average one-hop neighbor degree of all nodes in the network.

  Low: Average node degree of 8.

  Medium: Average node degree of 12.

  High: Average node degree of 16.

- **Node Mobility**: Introducing mobile *anchors* into a network has been shown to improve localization performance. However, introducing mobile *unknowns* into a network will likely increase the data communications and associated power costs of performing localization. Therefore, only mobile *unknowns* are modeled in the experiments to isolate the impact they have on the power required to localize. For these experiments, node mobility is modeled as a distinct random exit and entry point into the network. Additionally, once a node estimates its position, it remains static from that point on.

  None: All nodes are static.

  Low: At most, 10% of all *unknowns* are mobile at any given time.

  Medium: At most, 30% of all *unknowns* are mobile at any given time.

  High: At most, 50% of all *unknowns* are mobile at any given time.

3.9 Evaluation Technique

These experiments are conducted through simulation. The system is modeled, simulated, and analyzed using OPNET Modeler 10.5. Simulation is chosen over an-

alytical evaluation since no analytical models exist for measuring localization performance in AWSNs. Direct measurement evaluation is not practical given the cost, size, and complexity of setting up an operational AWSN. Evaluation using simulation is ideal as it allows for repeatable and controllable experiments. The experimental designs are validated by comparing the results to those found in the published research.

3.9.1 Experimental Setup. Both algorithms are designed and modeled using the MAC, MAC layer interface, and WLAN transmitter & receiver of the Wireless Station node model found in the OPNET 10.5A component library. Custom process models are added to perform the functions of the localization algorithm being modeled.

Sections 3.9.1.1 and 3.9.1.2 discuss design details and decisions made while implementing the two algorithms.

3.9.1.1 Map-Growing Implementation Details. A brief overview of the Map-Growing algorithm is provided in [LSS04]. However, the lower-level implementation details are not discussed in any detail. Therefore, many modeling and implementation decisions are made that may effect the performance of the algorithm. This section discusses several of the major decisions in the development of the Map-Growing algorithm modeled using OPNET 10.5A (MG-OP).

The Map-Growing algorithm consists of three main phases. The first phase establishes the center of a relative local map. In [LSS04], it is assumed all nodes already have distance estimates to all one-hop neighbors. To account for the communications cost of this, MG-OP has every node randomly transmit a ranging beacon during the first minute of simulation. A Gaussian ranging error is applied by the nodes receiving the beacon messages at this time. During the second minute of the simulation, nodes randomly send control messages to share and obtain all one-hop neighbor information, which includes degree and distance estimates.

The degree information is used to randomly select a single starting node that meets the requirement of having a degree that is greater than or equal to the degrees of all one-hop neighbors. To implement this in MG-OP, a flooding algorithm is used to coordinate the selection of a single starting node. Furthermore, MG-OP goes a step further and selects the node with the highest one-hop neighbor degree. If multiple nodes have the highest degree, then the one with the smaller source ID is selected. The flooding process begins by having all eligible nodes randomly transmit a starting message that advertises its degree and eligibility. Upon receiving an election message, a node compares the degree of the eligible node with the highest eligible degree known to that point, and takes one of the following three actions:

1. If the degree is greater, or if the degree is equal and has a smaller source-ID, the message is forwarded.
2. If the degree is less than or equal to the highest known degree with a larger source-ID, the message is deleted.
3. If the eligible node is currently winning the election, the message is deleted.

The flooding process continues until the end of the election time window. At this time, the elected starter identifies itself, if-and-only-if it did not receive a message from another eligible node with a higher degree or equal degree and smaller source-ID.

The starting node selects two one-hop neighbors to be part of the starting center island as shown in Figure 3.2. MG-OP selects the two neighbors with the highest sum degree that meet the collinearity requirements (all $\angle$'s $> 30°$). These three nodes establish a relative coordinate system with the starting node, u, at location (0,0) and the second node, v, on the relative X-axis at location (a,0). The third node, s, computes its relative (x,y) coordinates using (3.1) and (3.2) below [LSS04]. Note that (3.2) below corrects a misprint [LSS04]. Furthermore, MG-OP always chooses the

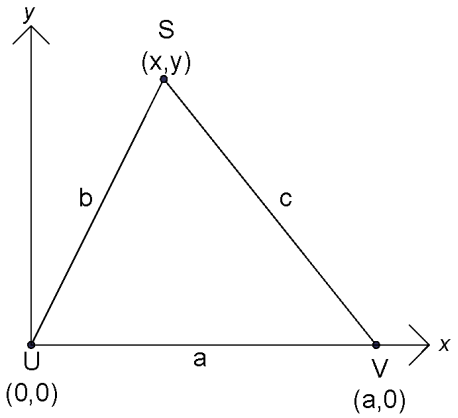

Figure 3.2: Map-Growing: Relative Map of Starting Center Island

positive value for y. The (x,y) coordinates are

$$x = \frac{a^2 + b^2 - c^2}{2a} \tag{3.1}$$

$$y = +\frac{\sqrt{(-a+b-c)(a-b-c)(a+b-c)(a+b+c)}}{2a} \tag{3.2}$$

where a is the distance from U to V, b is the distance from U to S, and c is the distance from S to V.

With the relative coordinates established for the center island, the Map-Growing phase begins. Upon localizing, a node broadcasts its estimated relative position. During this phase, all *unknown* nodes localize either using trilateration or a 2-*Anchor* localization method, depending on the relative *anchor* conditions shown in Figure 3.3.

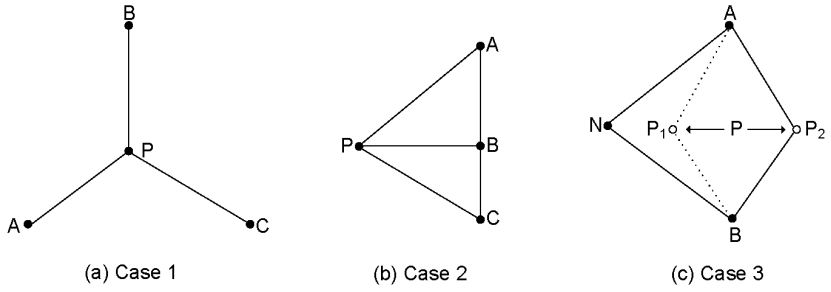

Figure 3.3: Map-Growing: Relative Anchor Conditions [LSS04]

Trilateration is always used when a node knows of three or more non-collinear relative *anchor* neighbors (Figure 3.3, Case 1). Appendix A provides an example of trilateration calculations. MG-OP localizes by averaging the trilateration results for every combination of the three relative *anchors* that meet the residue criteria. Residue is defined as the average of the differences of (1) the sum of the calculated distance estimates between nodes (using the position estimate), and (2) the sum of the original distance estimates (obtained in the ranging step). A large residue indicates an "inconsistent set of equations" or a more inaccurate position estimate [LR03]. In [LR03], a position estimate is rejected when

$$Residue = \frac{\sum_{i-1}^{n} \sqrt{(x_1 - x)^2 + (y_1 - y)^2} - d_i}{3} \leq Max_{Range}. \tag{3.3}$$

However, pilot studies show that an even smaller maximum residue criteria avoids extremely large position errors while not negatively affecting the algorithms ability to localize. Thus a lower residue threshold results in a lower average position error for the entire network. Therefore, MG-OP uses a maximum residue criteria of $\frac{Max_{Range}}{12}$.

Once a node receives a message from a second relative *anchor*, it waits 30 seconds before attempting to localize with the 2-*Anchor* localize method. If a node subsequently learns of localized non-collinear *anchors* during this time, it localizes using trilateration as described above. However, if three or more collinear anchors are available (Figure 3.3, Case 2), or if only two neighboring *anchors* are available (Figure 3.3, Case 3), the 2-*Anchor* localization method is used which requires two-hop neighbor information. Therefore, during this phase all messages not only consist of one-hop neighbor information, but also all two-hop neighbor information. Instead of requiring the transmission of two-hop neighbor information on-demand, two-hop neighbor information is always transmitted to simplify the implementation of the algorithm. This decision, therefore, models the worst case data transmission requirements for this algorithm.

The 2-*Anchor* localization method is similar to trilateration in that all combinations of two *anchors* are considered and the position estimation results of each combination are averaged together. Specific details of this method are in Appendix A. At the conclusion of the method, nodes subsequently use the "not being connected" constraint to choose between the two potential position estimates. This means at least one two-hop *anchor* is within maximum range of one but not both of the position estimates. Given it is within range of one estimate, that position is not plausible given the two-hop *anchor* would actually be a one-hope *anchor*. Therefore, the other position estimate is chosen.

In some cases, a pair of one-hop *anchors* know several neighboring *anchors* that are two-hop *anchors* of the localizing node. Additionally, in rarer cases there may be three or more collinear one-hop neighbors. In these cases, MG-OP performs the 2-*Anchor* localization method on all pairs of one-hop *anchors* and tests for the "not being connected" constraints on every localized two-hop *anchor*. MG-OP then estimates the position as the average of all valid results of every iteration of the 2-*Anchor* Localize method.

The third and last phase of the algorithm is the Transformation phase where relative coordinates are transformed into global coordinates. To perform this transformation, every node must obtain the relative and global coordinates of three or more non-collinear global *anchor* nodes by having all global *anchors* initiate a network flood, similar to the one used in the election of the starting node. When obtaining this information, all nodes with the exception of global *anchors* transform their relative coordinates into global ones using the affine transformation method described in Appendix A. As with lateration, the affine transformation is performed on all combinations of three global *anchors* and the estimated position results of each combination that meets the residue criteria are averaged together. However, the maximum residue criteria for the transformation is set to the maximum range, to transform a higher percentage of the nodes. So, although the smaller residue limit does not have a significant impact on the ability of nodes to localize during the relative localization step, it

does during the transformation. This is likely due to the increased distance separation and the associated accumulation in distance errors between the nodes localizing and the global *anchors*.

3.9.1.2 APS-Euclidean Implementation Details. A brief overview of the APS-Euclidean algorithm is provided in [NN01]. This section discusses some of the lower-level implementation details and choices made in the development of the APS-Euclidean algorithm modeled using OPNET 10.5A (E-OP).

The main computational complexity of APS-Euclidean is in the methods that derive distance estimates to *anchors* that may be multiple hops away. Distances to *anchors* are estimated one of two ways, either by voting between multiple combinations of one-hop neighbors or by examining the relationship between one-hop neighbors, two-hop neighbors, and *anchors*.

Figure 3.4 depicts the basic concept of APS-Euclidean's method for propagating *anchor* distances. Suppose the node trying to localize, *Self*, knows the distances a, b, c, d, and e, it is able to determine it is either at distance $r1$ or $r2$ away from the *Anchor*. E-OP determines the distances $r1$ and $r2$ the same way MG-OP establishes the relative positions of the center island. It then follows the two potential relative positions for *Self* in relation to $n1$, $n2$, and the *Anchor*, are computed using the 2-*Anchor* localize method. Given the relative local map, the distances are obtained using Pythagora's generalized theorem for calculating the distances between two known points.

The preferred method of localizing in APS-Euclidean is by one-hop neighbor voting. This method performs the above for two or more different pairs of one-hop neighbors that meet the criteria of (a) being neighbors with one another, (b) both have distance estimates to the same *anchor* and a common one-hop neighbor (other than *Self*). Additionally, the common one-hop neighbor must also have a distance estimate to the same *anchor*. Given multiple combinations of one-hop neighbors that meet this criteria, *Self* estimates its position as the average of half the distance estimates that are the same (given no ranging errors), or the set of half of the distances

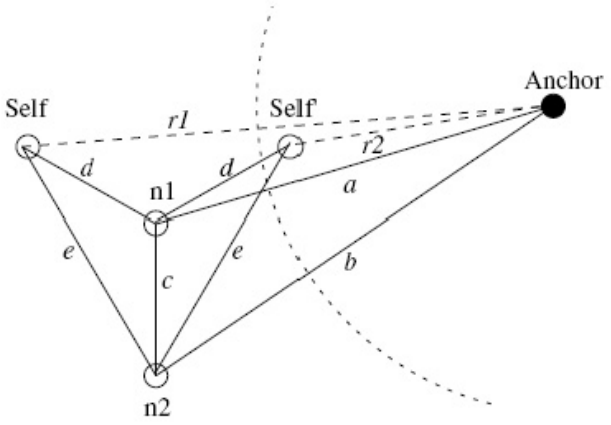

Figure 3.4: APS-Euclidean: Anchor Distance Propagation [LR03]

that have the smallest standard deviation (given ranging errors). [NN01] does not discuss setting a limit on the maximum number of combinations of one-hop neighbors, or their subsequent one-hop common neighbors for this voting method. Thus, E-OP does not restrict this number. Therefore, the computational complexity of E-OP is a worst case.

The secondary method for determining distance to an *anchor* consists of performing the voting method with a single pair of one-hop neighbors, followed by examining their relation with a common neighbor. This common neighbor must not be a one-hop neighbor of *Self* and it also must have a distance estimate to the *anchor* of interest. It follows that *Self* uses the "not being connected" constraint (also used in MG-OP), to choose between the two position estimates.

This requires *Self* to not only know the distance estimates between one and two-hop neighbors, but also the distance estimates from two-hop neighbors to *anchors*. Therefore, like MG-OP, this implementation of APS-Euclidean requires nodes to always forward one-hop and two-hop neighbor information.

Once a node obtains distance estimates to three or more *anchors*, it uses trilateration to estimate its position. Even after a node localizes, it continues to estimate distances to additional anchors to further propagate *anchor* distances that may assist

the rest of the network in localizing. No limit is placed on the maximum number of *anchors* a node will use to determine a distance estimate.

3.10 Experimental Design

A full factorial design is conducted for this experiment. Two of the factors have 2 levels, one has 3, and the other has 4 levels. Therefore, a single replication of the full factorial design consists of $2\times2\times3\times4 = 48$ experiments. These experiments are shown in Table 3.2. The advantage of this design is that every possible combination of configuration is examined, allowing the effects of every factor and their interactions to be determined.

Table 3.2: Experimental Design: Factors & Levels

Localization Algorithm	Network Shape	Network Degree	Node Mobility
APS-Euclidean	Full Square C-Shaped	L, M, H L, M, H	None, L, M, H None, L, M, H
Map-Growing	Full Square C-Shaped	L, M, H L, M, H	None, L, M, H None, L, M, H

The length of each simulation varies according to the time required for each experiment to successfully converge on position estimations. Using a 90% confidence interval, repeating each experiment 30 times provides a sufficient statistical basis for analysis. The largest variance in the data will most likely occur in the experiments with the most irregularity (e.g., C-Shaped, Low Degree, and High Mobility) since network irregularity introduces significantly more variability into the experiment. Given 30 repetitions, a total of 1,440 simulations are executed. Assumptions about the data and measurements errors are (1) measurement errors are statistically independent, (2) measurement errors are normally distributed with a mean of zero, and (3) the variance of measurement errors is constant. These assumptions are verified using the appropriate approximate visual tests.

3.11 Summary

This chapter defines a methodology to determine the effect of network shape, degree, and node mobility on the data communications and associated power needed for a node to localize nodes in an AWSN. The system boundaries, services, and workload are defined. Additionally, performance metrics, system parameters, and experimental factors are explained in detail. Lastly, the evaluation technique and experimental design is given.

IV. Results and Analysis

4.1 Introduction

This chapter presents experimental results and analysis. Section 4.2 discusses the validation results of the OPNET implementations of the Map-Growing and APS-Euclidean localization algorithms. Section 4.3 explains how the data is collected and analyzed. Section 4.4 evaluates the results of the Mean Percent Localized and Mean Position Error Responses. Section 4.5 presents an in-depth analysis of the Bits Transmitted and Bits Received responses for both algorithms. Section 4.6 presents a power model for each algorithm based on a multiple linear regression of the Bits Transmitted and Bit Received responses. Section 4.7 briefly discusses other findings of this research. Lastly, Section 4.8 provides a summary of the chapter.

4.2 Algorithm Validation

The OPNET implementation of both algorithms are validated by comparing the simulation results for percent localized and position error to those published in [LSS04] and [NN01] respectively. The ultimate goal is to determine if the OPNET implementations result in similar behavior and performance.

4.2.1 Map-Growing Algorithm. The OPNET implementation of the Map-Growing algorithm (MG-OP) is tested on both the Normal Grid and the Normal Random network layouts described in [LSS04]. Both scenarios are run with 13 different range errors, from 0% to 12%. Thirty repetitions of the 26 experiments are run, resulting in a total of 780 simulation runs. Table 4.1 shows the parameter values for both the Normal Grid and Normal Random scenarios.

Table 4.1: MG-OP: Validation Simulation Settings

Validation Parameter	Normal Grid Settings	Normal Random Settings
Square Side Length (units)	18	20
Number of Nodes	100	200
Range (units)	3	3
Average Node Degree	6.5	12.3

The published results state that 100% of nodes are localized for every scenario. However, the simulated results ranged from an average of 98% to 100% of nodes localized. Inspection of several of the specific trials where 100% localization is not achieved shows that nodes not localized are either fully disconnected, partially connected (1-neighbor), or minimally connected (2 or 3 neighbors) with the inability to meet localization requirements due to near-collinear conditions combined with large range errors. Since the Map-Growing algorithm does not clearly define techniques allowing for successful localization in these situations, the slightly lower percent localized results are deemed unavoidable.

The simulation results for the Normal Grid scenario are shown in Figure 4.1 and the results for the Normal Random scenario are shown in Figure 4.2. The published results [LSS04] are visually estimated and shown as the "basis" data series in both figures. Comparing the simulated results (with 90% confidence intervals) to the "basis" clearly shows MG-OP does not achieve the same position error performance. It

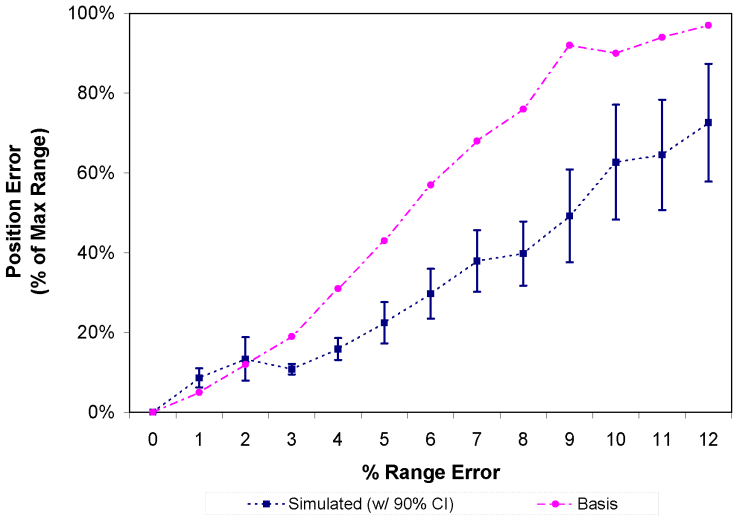

Figure 4.1: MG-OP: Mean Position Error Response for Normal Grid Network

is interesting to note that the average position error performance of MG-OP is generally better for the Normal Grid scenario but worse for the Normal Random scenario. The published results [LSS04] state that 900 repetitions are performed. However,

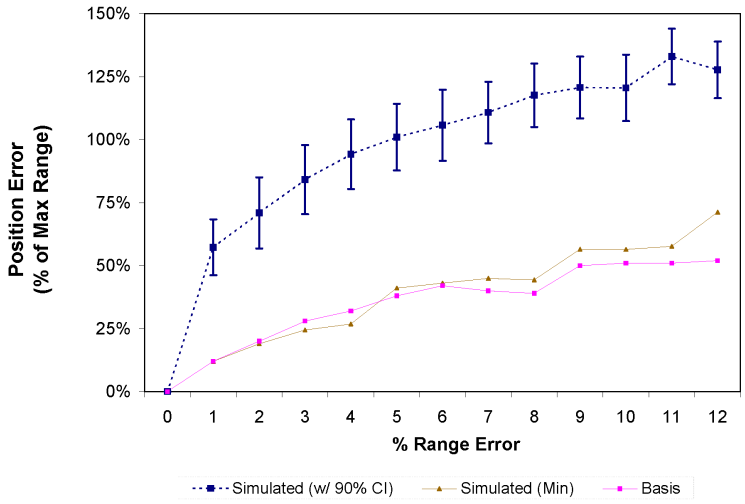

Figure 4.2: MG-OP: Mean Position Error Response for Normal Random Network

no confidence intervals are given and the raw data was unavailable when requested. Although, the average position error results are much higher, the best simulated experiments (the "Simulated (Min)" data series) achieved a position error very close to the "basis". All experiments with zero range error result in no error in the position estimation.

Since many of the lower-level implementation details of the Map-Growing algorithm are not provided, it is possible that differences and missing optimizations may be the cause of the performance differences. However, it is also important to mention that "basis" data was the result of a Matlab simulation. Since Matlab is a table-driven simulator and OPNET is an event-driven simulator, this may also contribute to the performance differences. In the OPNET simulation, nodes receive update messages from neighbors at varying times throughout the scenario. These update messages often only contain partial neighborhood information, resulting in nodes making localization decisions without having all of one-hop and two-hop node information. It is possible that the Matlab simulation assumes total awareness of this information when making decisions. Other than these conjectures, no reason for the performance difference could be determined. However, the trends of the OPNET simulations are

similar to the "basis". Furthermore, because position error is not used a basis for comparison, the model is deemed a good representation of an anchor-based incremental localization algorithm and is therefore acceptable for the purpose of this research.

4.2.2 APS-Euclidean Algorithm. The OPNET implementation of the APS-Euclidean algorithm (E-OP) is tested using a normal random network scenario [NN01]. Table 4.2 lists the parameters values used and the factor levels varied for these experiments. How *anchors* are placed in the network are not mentioned in [NN01]. Therefore, *anchors* are uniformly selected from the random uniform distribution of nodes. A full-factorial experiment is performed on 5 different random network layouts, resulting in 180 simulations.

Table 4.2: E-OP: Simulation Settings

Parameter / Factor	Settings
Square Side Length (units)	17.5
Number of Nodes (units)	100
Range (units)	3
Average Node Degree	7.6
Anchor Ratio	0.05, 0.1, 0.2, 0.5, 0.9
Percent Range Error (StDev)	0.0, 0.02, 0.05, 0.1, 0.2, 0.5, 0.9

The simulation results for the mean percent localized response are shown in Figure 4.3, while the published percent localized results are shown in Figure 4.4. The results are close to the published results and generally follow the same trend and therefore are acceptable.

The simulated results for the mean position error response are shown in Figure 4.5, while the published results are shown in Figure 4.6. Generally, E-OP follows the same performance trend of the published results. There are only four points in the simulated results that bear mentioning. Two configurations, 10% anchors with 10% range errors, and 10% anchors with 90% range errors, were approximately 20% below the published results. Also, the results of the configuration with 20% anchors

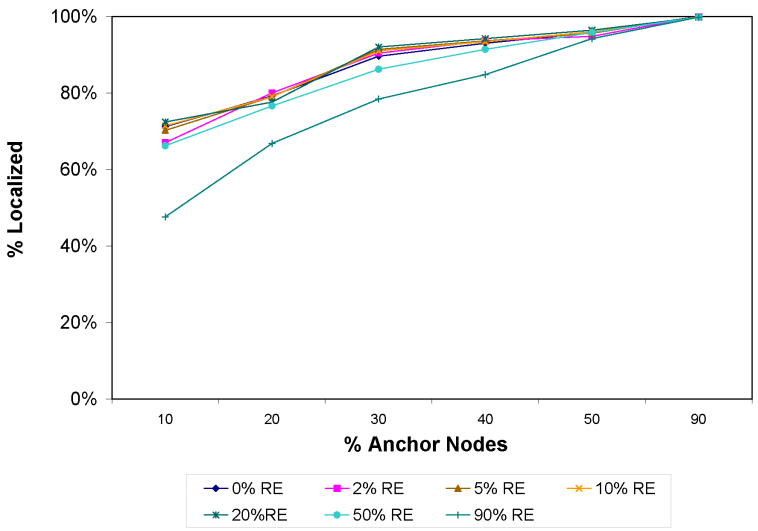

Figure 4.3: E-OP: Simulated Percent Localized Response

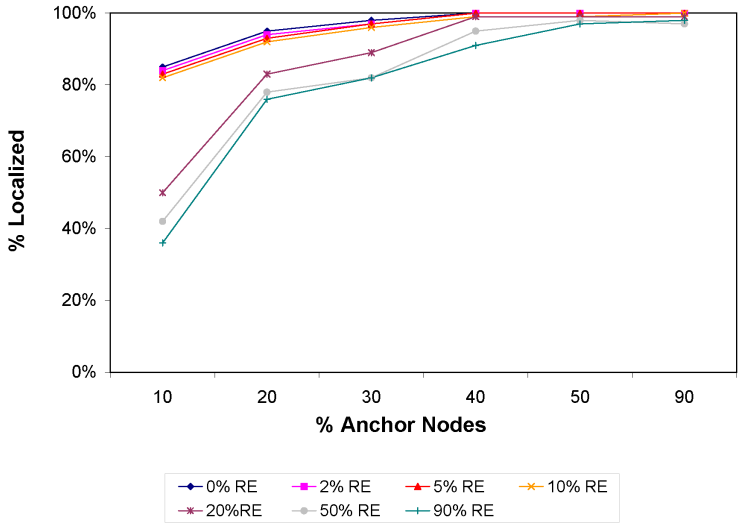

Figure 4.4: APS-Euclidean: Previously Published Percent Localized Response [NN01]

and 20% range errors are about 15% below the published results. Lastly, the 90% range error with 90% anchors is approximately 10% above the published results. All other design points are within about ±5% of the published results and all experiments with zero range error resulted in no error in the position estimation. This leads to the conclusion that E-OP performs similarly to the published results and is functionally sound. The model is a good representation of a anchor-based concurrent localization

algorithm, since position error is not used as a basis for comparison and the trends are similar. Therefore it is acceptable for the purpose of this research.

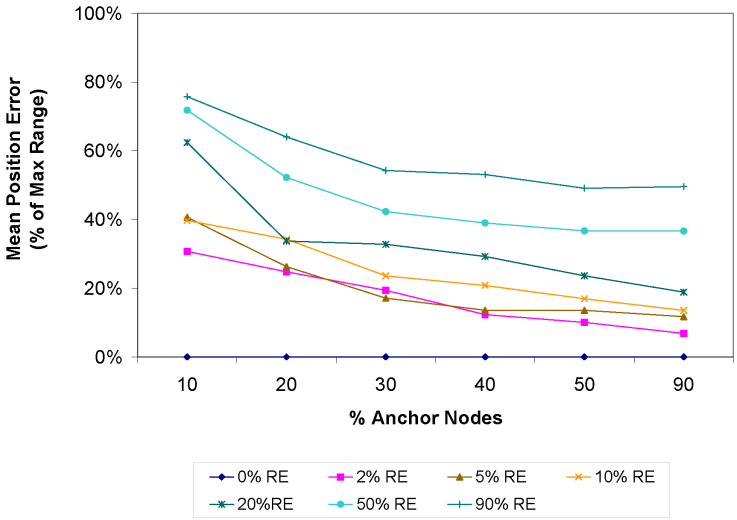

Figure 4.5: E-OP: Simulated Mean Position Error Response

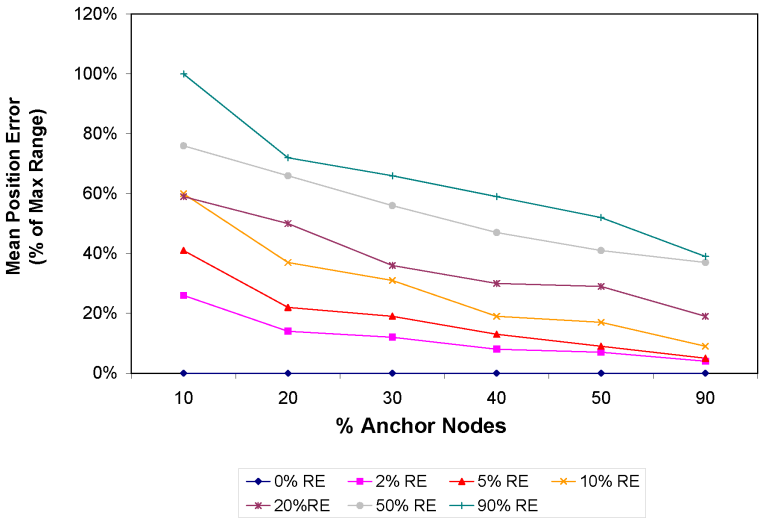

Figure 4.6: APS-Euclidean: Mean Position Error Basis [NN01]

4.3 Data Collection and Analysis Methods

The experiments are tested or repeated on 30 different random network layouts. The length of the experiments are determined by the time it takes to reach 98%

localization. However, if 98% localization is not achieved by a certain point, then the experiment ends regardless. For MG-OP, this time is 2,700 seconds and for E-OP 1,700 seconds. MG-OP requires an extra 1,000 seconds to complete the election and transformation phases. Additionally, both algorithms have a 1,600 second window in which nodes may be mobile. For MG-OP, mobility starts at approximately 400 seconds, immediately following the election phase and ends at 2,000 seconds. However, E-OP initiates mobility at the beginning of the simulation and ends at 1,600 seconds. For both algorithms, 100 seconds is allocated after the end of mobility, to allow newly positioned 'mobile' nodes to localize. After the 100 second window, MG-OP has 600 seconds to complete its transformation phase.

Data Transmissions and Receives are calculated in the OPNET custom process models and a running total is maintained by each individual node. Only the size of the data being sent is recorded. Therefore, no lower-level communications overhead such as packet headers and trailers are considered. However, additional overhead can be later accounted for as a percentage of goodput in the regression equations. Given that the amount of data sent is determined by the type of message and the amount of neighbor information being sent, messages vary in size. Upon completion of a trial, each node computes the totals for each response and records the results in a file. Furthermore, at the end of every run the appropriate responses for the entire network are calculated and saved to a separate file.

The simulation generates individual values for position error, bits transmitted and bits received for every node. The simulation also records the mean value for position error and percent localized, the total number of bits transmitted and received. MiniTab 14 is used to further consolidate the data and display it in graphical form. The computation of effects are computed in Microsoft Excel, and the Analysis of Variance (ANOVA) for the performance metrics are performed with MiniTab 14. MiniTab 14 provides native support for n-way ANOVA computations. All insignificant factors (P-value > 0.1) and related higher-order interactions are removed (pooled) from all ANOVA calculations. The result of pooling is that the degrees of freedom

and the sum of squares for all removed factors and interactions are added to the degrees of freedom and sum of squares of the error term, and mean square terms are recalculated. The ANOVA tables presented reflect the final calculations after pooling.

After implementing node mobility into the custom process models, a simulation abort due to invalid memory access occurred randomly for about 10% of the network configurations. The total number of aborts for any one particular configuration is approximately 33%. The cause of this problem was not determined. However, all of the completed trials appear to behave normally. Therefore, 45 different random seeds are used for the MG-OP experiments, of which 30 of the successfully completed repetitions are randomly selected and used. Similarly, 40 different random seeds are used for E-OP, of which 30 are also selected at random.

4.4 Percent Localized & Position Error Response Analysis

The main objective of an AWSN localization algorithm is to localize every node with the smallest achievable position error. This section presents the experimental results and analysis for the percent localized and position error performance metrics.

Figure 4.7 shows the mean percent localized and mean position error responses (with 90% confidence intervals) versus the algorithm and network shape factors. As expected, the plot shows that both algorithms localize fewer nodes for C-shaped networks than they do for full Square networks. However, the resulting decrease in percent localized performance in C-Shape networks is less drastic for E-OP. Figure 4.7 also shows that E-OP results in a mean position error performance significantly lower than MG-OP with 90% confidence, regardless of the network shape.

Similarly, Figure 4.8 illustrates the effects of algorithm and degree on the mean percent localized and mean position error responses. Both algorithms result in the same trend. That is, as the degree of the network increases, so does the percentage of nodes able to localize. This confirms the intuition that the higher the average degree of a network, the easier it is for a localization algorithm to converge. It is also

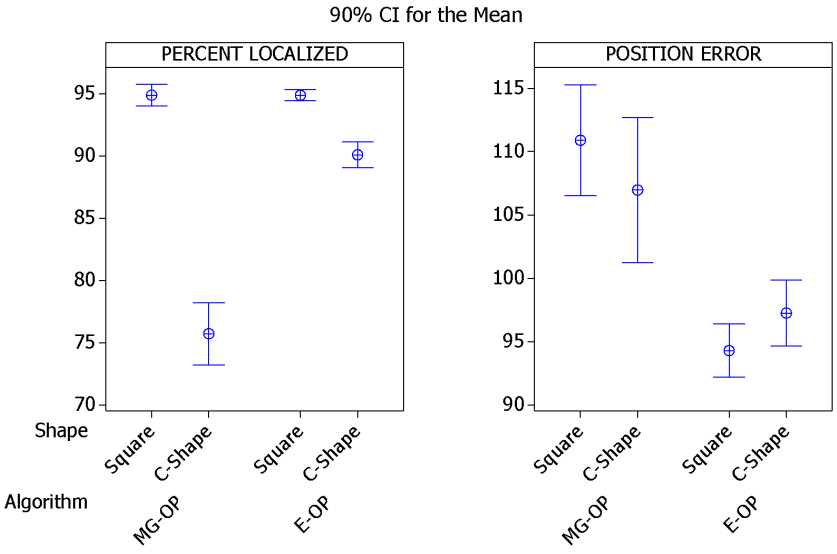

Figure 4.7: Interval Plots: Mean Percent Localized and Position Error Responses versus Algorithm and Shape

interesting to note how the reduction in percent localized between degree 12 to 8 is substantially more than between degree 16 to 12. Specifically, the 68% localization achieved by MG-OP, indicates that low degree networks are unsuitable for the Map-Growing algorithm.

The low percent localized performance associated with low degree networks, has a direct impact on the corresponding mean position error; drastically lowering the mean position error performance for MG-OP degree 8 networks. However, this is a reasonable response since networks with significantly fewer nodes localizing (holding other factors constant) have much less propagation and accumulation of ranging/positioning error. In principle, the closer (in hop counts) localizing nodes are to *anchors*, the more accurate their position estimate. For the incremental Map-Growing algorithm especially, as a higher percentage of nodes localize, the overall average position error increases due to the larger position error results of nodes further away from the center island. With the exception of MG-OP degree 8, the position error significantly decreases as the degree level increases, at a $\alpha = 0.1$ significance level. This confirms the expected trend that a higher degree results in more accurate position estimates.

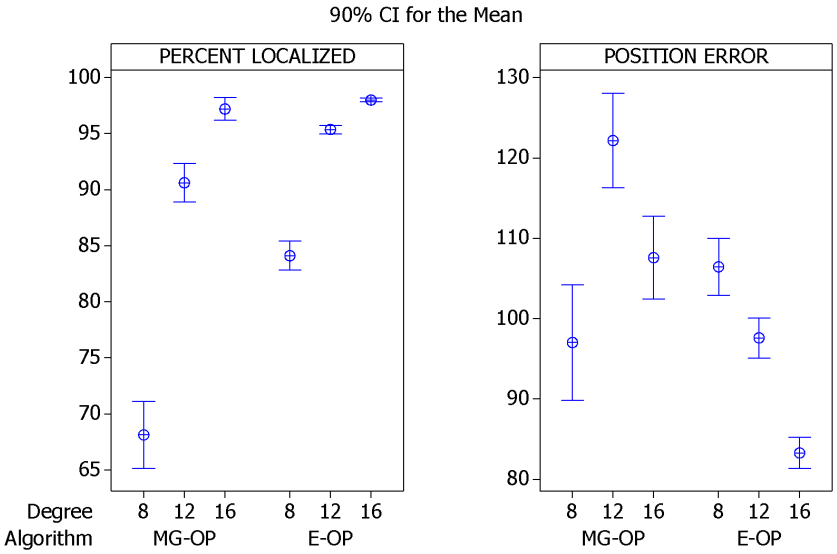

Figure 4.8: Interval Plots: Mean Percent Localized and Position Error Responses versus Algorithm and Degree

The interval plot also shows that E-OP achieves significantly lower mean position error than MG-OP for networks with corresponding degrees.

Figure 4.9 show the interval plots for percent localized and mean position error responses versus algorithm and mobility. The most obvious conclusion is 10% and 30% mobility levels significantly improve the percent localized response with 90% confidence, regardless of the algorithm. Intuitively this effect makes sense, since node mobility allows unconnected or poorly positioned nodes to potentially move to other locations with better connectivity. Also, with the exception of the 10% mobile networks, E-OP localizes significantly more nodes than does MG-OP. It is likely the inability for MG-OP to localize with high mobility is a direct result the incremental approach requiring a stable and uniformly distributed network to grow in a timely manner. These results indicate that high mobility networks are unsuitable for the Map-Growing algorithm. It is interesting to note how the percent localized and position error plots share the same trends. This adds additional support to the fact that the mean position error response is directly related to the percent localized response.

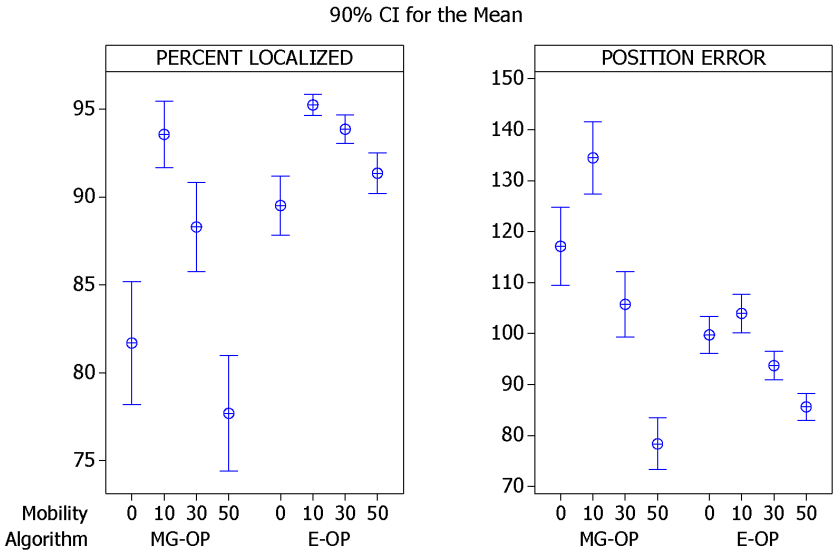

Figure 4.9: Interval Plots: Mean Percent Localized and Position Error Responses versus Algorithm and Mobility

Figure 4.10 shows the mean percent localized and mean position error for both algorithms. Since the confidence intervals don't overlap, the differences in the mean percent localized and mean position error responses of the two algorithms are statistically significant at a $\alpha = 0.1$ significance level. This indicates E-OP localizes more nodes and achieves more precise position estimates than does MG-OP.

4.5 Communications Response Analysis

This section presents an in-depth analysis, to include computation of effects and an Analysis of Variance (ANOVA) for the total Bits Transmitted and total Bits Received responses. The Bits Transmitted and Received responses are measured as the sum of all data bits transmitted and received by every node in a network. For readability of the graphs, 1,000,000 bits equals 1.0 Mega-bits.

Prior to statistical analysis of the data, visual tests confirm the ANOVA assumptions of independent observations and normally distributed residuals with constant variance. Initial visual tests for both of the responses show the following:

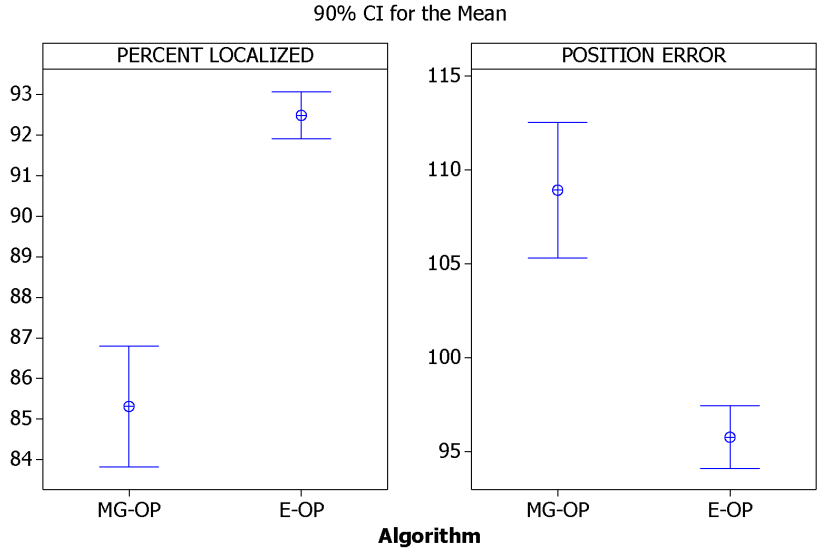

Figure 4.10: Interval Plots: Mean Percent Localized and Position Error Responses versus Algorithm

- The scatter plot of Residuals versus Fitted Values show an increasing trend, indicating a dependence of errors on the factor levels and a non-constant variance of errors.
- The Normality Plot of Residuals is non-linear (s-shaped), indicating residuals are not normally distributed.
- The ratio $\frac{y_{max}}{y_{min}}$ is large for each of the responses, indicating a transformation is most likely necessary [Jai91].

Therefore, since observations are not independent and normally distributed with constant variance, a Box-Cox (BC) Transformation is performed on both responses [Jai91]. MiniTab 14 determined the best BC transformation had parameter a between 0.3 and 0.4. Therefore, $a = 0.4$ is used for both responses in both algorithms. The visual tests on the transformed data show no trends and a constant spread in the Residuals versus Fitted Values. The Normal Plot of Residuals is reasonably close to a straight line. However, for MG-OP, the Normal Plot of Residuals result in short tails consisting of approximately 50 outliers among the two responses. These outliers are

shown in the Normal Plot of Residuals for the BC-Bits Received response shown in Figure 4.11. Inspection of the outliers determined they correspond to the high mobility and/or low degree networks. These outliers are likely the result of uncontrollable variability introduced by these factor levels. Thus, we propose that low degree and high mobility networks may not be suitable for the Map-growing algorithm given this instability, and the outliers are subsequently removed. The resulting Normal Plot of Residuals shown in Figure 4.12 shows a good linear fit of residuals. Therefore, the assumptions the transformed data is independent and normally distributed with a constant variance are met. The results of all visual tests used for validating the ANOVA assumptions are provided in Appendix C.

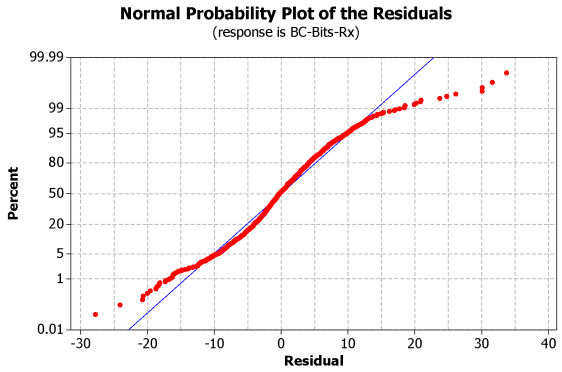

Figure 4.11: Normal Plot of Residuals for BC-Bits Received Response (with outliers)

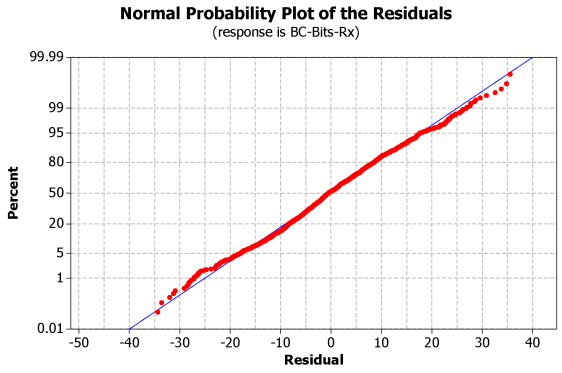

Figure 4.12: Normal Plot of Residuals for BC-Bits Received Response (outliers removed)

The largest number of high mobility, low degree outliers in a single experimental configuration is 8. Therefore, to maintain a balanced experimental design, the number of each MG-OP configuration is reduced by 8 by randomly removing non-outliers from the results as necessary. This reduces the number of observations for MG-OP to 22. The number of APS-Euclidean observations remain at 30. All statistical analysis presented is based on the remaining observations, with all identified outliers removed.

Since the number of bits received is a function of the total number of bits transmitted, both responses typically follow the same trends. Therefore the analysis of these two responses is presented together. Sections 4.5.1, 4.5.2, and 4.5.3 present the results and analyze how network shape, node degree, and node mobility affect them. Section 4.5.4 presents the computation of effects on the responses for both algorithms, while Section 4.5.5 provides the corresponding Analysis of Variance for the responses.

4.5.1 Shape. Figure 4.13 shows the main effect of a C-Shape networks is a decrease in the number of transmitted and received bits. That is, the number of bits transmitted and received in a C-Shape network is significantly less than the Square networks, at a $\alpha = 0.1$ significance level. The one exception is with E-OP where the difference between the bits received for the two shapes is not significant at $\alpha = 0.1$ significance level. These results do not support the hypothesis that network irregularity leads to an increase in communications costs. From the previous section, it is apparent that the irregular network shape ultimately affects the ability of nodes to localize. This inability to localize results in a decrease in the amount of data known about the network; ultimately decreasing the communications cost of the network. However, it is likely that if the percent localized response were comparable to that of a Square network, the C-Shape network would require a comparable, and most likely not significantly different, number of bits transmitted and received.

4.5.2 Degree. The interval plots in Figure 4.14 confirm that an increasing average node degree results in a higher communications cost. Although the responses

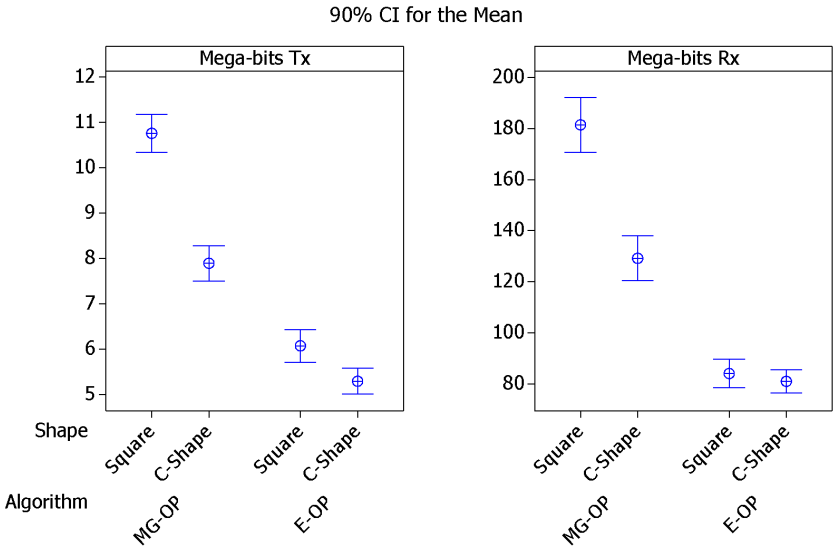

Figure 4.13: Interval Plot: Communications Responses versus Algorithm and Shape

of both algorithms support this fact, the most apparent effect is in the performance of MG-OP. This is expected, since the amount of neighbor data sent increases with neighbor degree. A visual comparison of the confidence intervals for the E-OP degree 12 and 16 networks show that the differences in the Bits Transmitted responses are not statistically significant at a $\alpha = 0.1$ significance level. This indicates degree has less of an effect on communications for the E-OP algorithm than for MG-OP.

4.5.3 Mobility. The interval plots shown in Figure 4.15 confirm an increase in the percentage of mobile nodes increases the number of bits transmitted and received during localization. For both algorithms and both responses, the difference in the mean responses from one level of mobility to another is statistically significant at a $\alpha = 0.1$ significance level.

4.5.4 Computation of Effects. This section analyzes the effects of each level of the shape, degree, and mobility factors. The effects of the factor levels on the Mega-bits Transmitted and Mega-bits Received responses are analyzed separately

for each algorithm. Supporting tables for the computation of effects are provided in Appendix B.

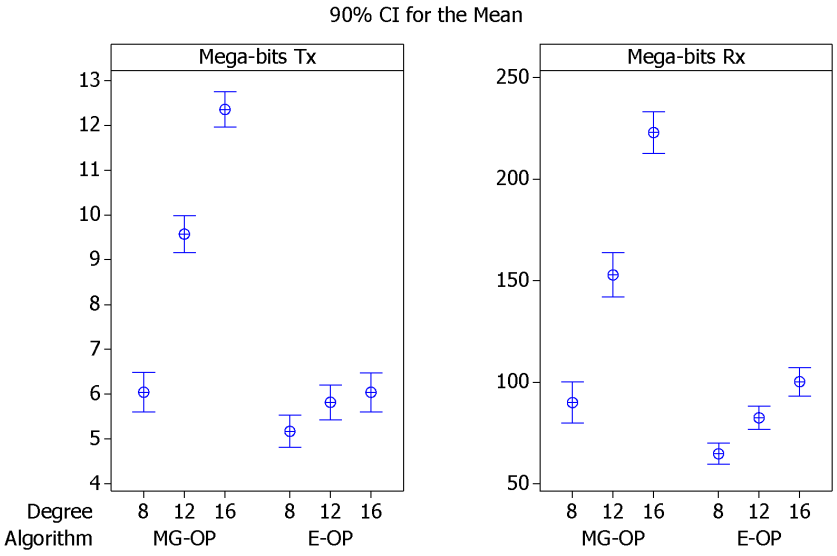

Figure 4.14: Interval Plot: Communications Responses versus Algorithm and Degree

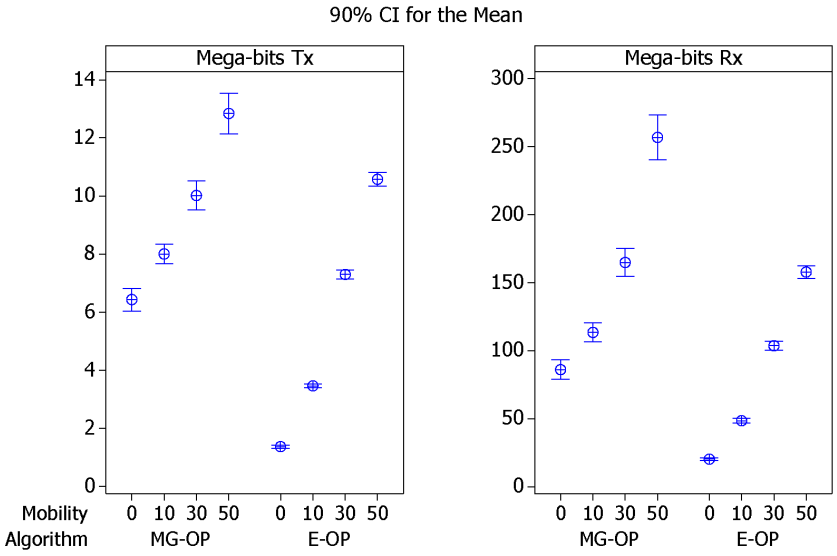

Figure 4.15: Interval Plot: Communications Responses versus Algorithm and Mobility

4.5.4.1 Bits Transmitted.

MG-OP. A summary of results for the computation of effects on the Mega-bits Transmitted response for MG-OP is shown in Table 4.3. The corresponding computation of effects tables are provided in Appendix B. The overall mean response is 9.32 Mega-bits Transmitted. Since zero is not included in any of the confidence intervals, the effects of all levels of all factors are statistically significant at an $\alpha = 0.1$ significance level. Furthermore, since no confidence intervals overlap for the levels of any particular factor, all effects of the levels are significantly different from one another with 90% confidence. As is the case with MG-OP, mobility and degree also have the largest effect on the Bits Transmitted response for E-OP. Specifically, the mean response of the 50% mobility level is approximately 37% higher than the mean, and 100% higher than the mean response of a static network. Similarly, the mean response for a high degree network is approximately 34% higher than the mean, and 104% higher than a low degree network.

Table 4.3: MG-OP: 90% Confidence Intervals for Main Effects on Mega-bits Transmitted

Parameter	Mean Effect	Std Dev	90% Confidence Interval
Mean	9.32	0.05	(9.24, 9.41)
Mobility Effects			
Static	-2.89	0.09	(-3.03, -2.75)
10% Mobile	-1.32	0.09	(-1.46, -1.18)
30% Mobile	0.69	0.09	(0.55, 0.83)
50% Mobile	3.52	0.09	(3.38, 3.66)
Degree Effects			
Low	-3.28	0.07	(-3.40, -3.16)
Med	0.25	0.07	(0.13, 0.37)
High	3.03	0.07	(2.92, 3.15)
Shape Effects			
Square	1.43	0.05	(1.35, 1.52)
C-Shaped	-1.43	0.05	(-1.52, -1.35)

E-OP. A summary of results for the computation of effects on the Mega-bits Transmitted response for E-OP is shown in Table 4.4. This table shows the overall mean Mega-bits Transmitted response is 5.68. All factor levels have

effects that are statistically significant at an $\alpha = 0.1$ significance level. Additionally, the confidence intervals do not overlap so all levels of each factor are statistically different from one another. For E-OP, mobility has the largest affect on the number of bits transmitted, with the mean response of the 50% mobility level being approximately 86% higher than the mean, and 672% higher than the mean response of a static network.

Table 4.4: E-OP: 90% Confidence Intervals for Main Effects on Mega-bits Transmitted

Parameter	Mean Effect	Std Dev	90% Confidence Interval
Mean	5.68	0.03	(5.63, 5.73)
Mobility Effects			
Static	-4.31	0.05	(-4.39, -4.22)
10% Mobile	-2.21	0.05	(-2.30, -2.13)
30% Mobile	1.62	0.05	(1.53, 1.71)
50% Mobile	4.90	0.05	(4.81, 4.99)
Degree Effects			
Low	-0.51	0.04	(-0.58, -0.43)
Med	0.14	0.04	(0.07, 0.21)
High	0.36	0.04	(0.29, 0.44)
Shape Effects			
Square	0.39	0.03	(0.34, 0.44)
C-Shaped	-0.39	0.03	(-0.44, -0.34)

4.5.4.2 Bits Received.

MG-OP. A summary of results for the computation of effects on the Mega-bits Received response for MG-OP is shown in Table 4.5. The overall mean response is 155.28 Mega-bits. The effect of the medium degree level is not statistically significant at an $\alpha = 0.1$ significance level, since the confidence interval for the mean effect includes zero. All other effects are statistically significant at a $\alpha = 0.1$ significance level. Additionally, all effects are significantly different from one another with 90% confidence. The response follows the same trend as the Mega-bits Transmitted response does for this algorithm. Thus, mobility and degree have the largest affect on the mean response. The effects of the 50% mobile level is the most significant with a response that is 101.57 Mega-bits more than the mean response.

Table 4.5: MG-OP: 90% Confidence Intervals for Main Effects on Mega-bits Received

Parameter	Mean Effect	Std Dev	90% Confidence Interval
Mean	155.28	1.38	(153.01, 157.55)
Mobility Effects			
Static	-69.24	2.39	(-73.17, -65.31)
10% Mobile	-41.90	2.39	(-45.82, -37.97)
30% Mobile	9.56	2.39	(5.64, 13.49)
50% Mobile	101.57	2.39	(97.64, 105.50)
Degree Effects			
Low	-65.25	1.95	(-68.45, -62.04)
Med	-2.43	1.95	(-5.64, 0.77)
High	67.68	1.95	(64.47, 70.89)
Shape Effects			
Square	26.10	1.38	(23.83, 28.37)
C-Shaped	-26.10	1.38	(-28.37, -23.83)

Table 4.6: E-OP: 90% Confidence Intervals for Main Effects on Mega-bits Received

Parameter	Mean Effect	Std Dev	90% Confidence Interval
Mean	82.54	0.62	(81.52, 83.56)
Mobility Effects			
Static	-62.33	1.07	(-64.10, -60.56)
10% Mobile	-34.04	1.07	(-35.81, -32.28)
30% Mobile	21.09	1.07	(19.32, 22.86)
50% Mobile	75.29	1.07	(73.52, 77.05)
Degree Effects			
Low	-17.72	0.88	(-19.16, -16.27)
Med	-0.03	0.88	(-1.47, 1.42)
High	17.74	0.88	(16.30, 19.19)
Shape Effects			
Square	1.53	0.62	(0.51, 2.55)
C-Shaped	-1.53	0.62	(-2.55, -0.51)

E-OP. Table 4.6 summarizes the main effects on the Mega-bits Received response for E-OP. The overall mean response is 82.54 Mega-bits Received. All effects are significantly different from one another with 90% confidence. The effect of the medium degree level is not statistically significant at an $\alpha = 0.1$ significance level, given the confidence interval includes zero. All other effects are statistically significant at a $\alpha = 0.1$ significance level. The response follows the same trend as the Mega-bits Transmitted response does for this algorithm.

4.5.5 ANOVA. This section provides an Analysis of Variance of the predictors, shape, degree, mobility, and their respective interactions. The resulting proportion of variation in the response explained by each predictor is discussed. The Adjusted Coefficient of Determination, R^2-Adj, indicates the proportion of variation in the response data that can be attributed or explained by the factors. All predictors with P-values less than 0.1 are statistically significant with 90% confidence.

4.5.5.1 Bits Transmitted.

MG-OP. Table 4.7 is the ANOVA of the MG-OP BC-Bits Transmitted response. The coefficient of determination, R^2-Adj, indicates 92.85% of the variation in the response is explained by the factors and their interactions. The ANOVA results indicate 42.2% of the variation is attributed to degree, while only 27.6% of the variation is attributed to mobility. This is somewhat unexpected, given the amount of additional messages random mobility requires. However, it is likely that the election and transformation phases of the Map-Growing algorithm is essentially masking the effects of mobility. That is, all networks regardless of the mobility level participate in the election and transformation phases, which like mobility also requires a large amount of message traffic. Therefore, the degree of the network affects the bits transmitted during these stages significantly more than mobile nodes do during the mobility phase. Network shape explains 12.36% of the variation in the number of bits transmitted. Lastly, the interactions of the factors account for 11.01% of the variation, while the remaining 6.84% is attributed to experimental error.

E-OP. Table 4.8 shows the ANOVA of the BC-Bits Transmitted response using E-OP. The R^2-Adj indicates 96.23% of the variation in the response is explained by the factors and their interactions. The ANOVA indicates 92.97% of the variation is attributed to mobility, while only 0.85% by degree, and 0.59% by shape. This is not surprising given the number of messages introduced

Table 4.7: MG-OP: ANOVA for BC-Bits Transmitted

Component	Sum of Squares	Percentage Variation	Degrees of Freedom	Mean Squares	F-Computed	p-Value
SSY	1.95E+08		528			
SS0	1.88E+08		1			
SST	7.36E+06		527			
Shape	9.10E+05	12.36%	1	9.10E+05	911.15	< 0.0005
Degree	3.10E+06	42.20%	2	1.55E+06	1555.01	< 0.0005
Mobility	2.03E+06	27.60%	3	6.77E+05	678.04	< 0.0005
Shape*Degree	4.38E+05	5.96%	2	2.19E+05	219.45	< 0.0005
Shape*Mobility	2.82E+05	3.83%	3	9.39E+04	94.06	< 0.0005
Degree*Mobility	5.87E+04	0.80%	6	9.78E+03	9.79	< 0.0005
Shape*Degree*Mobility	3.08E+04	0.42%	6	5.14E+03	5.15	< 0.0005
Errors	5.03E+05	6.84%	504	9.98E+02		
	$S_e = 31.595$		R^2-Adj = 92.85%			

by random mobility. The interactions of the factors only account for 1.94% of the variation, while the remaining 3.65% is attributed to experimental error.

Table 4.8: E-OP: ANOVA for BC-Bits Transmitted

Component	Sum of Squares	Percentage Variation	Degrees of Freedom	Mean Squares	F-Computed	p-Value
SSY	1.38E+08		720			
SS0	1.24E+08		1			
SST	1.47E+07		719			
Shape	8.69E+04	0.59%	1	8.69E+04	112.45	< 0.0005
Degree	1.25E+05	0.85%	2	6.25E+04	80.83	< 0.0005
Mobility	1.37E+07	92.97%	3	4.57E+06	5911.17	< 0.0005
Shape*Degree	5.74E+04	0.39%	2	2.87E+04	37.14	< 0.0005
Shape*Mobility	1.05E+05	0.71%	3	3.51E+04	45.46	< 0.0005
Degree*Mobility	5.77E+04	0.39%	6	9.61E+03	12.43	< 0.0005
Shape*Degree*Mobility	6.61E+04	0.45%	6	1.10E+04	14.25	< 0.0005
Errors	5.38E+05	3.65%	696	7.73E+02		
	$S_e = 27.8$		R^2-Adj = 96.23%			

4.5.5.2 Bits Received.

MG-OP. Table 4.9 shows the ANOVA for the BC-Bits Received response using MG-OP. The R^2-Adj indicates 92.6% of the variation in the response is explained by the factors and their interactions. According to the ANOVA, degree and mobility explain 38.07% and 38.06% of the variation in the

response respectively. Network shape only account for 7.52% of the variation in the response. The interactions of the factors only account for 9.27% of the variation, while the remaining 7.08% is due to experimental error.

Table 4.9: MG-OP: ANOVA for BC-Bits Received

Component	Sum of Squares	Percentage Variation	Degrees of Freedom	Mean Squares	F-Computed	p-Value
SSY	1.82E+09		528			
SS0	1.69E+09		1			
SST	1.31E+08		527			
Shape	9.85E+06	7.52%	1	9.85E+06	535.42	< 0.0005
Degree	4.99E+07	38.07%	2	2.49E+07	1355.43	< 0.0005
Mobility	4.99E+07	38.06%	3	1.66E+07	903.38	< 0.0005
Shape*Degree	5.34E+06	4.07%	2	2.67E+06	145.02	< 0.0005
Shape*Mobility	4.89E+06	3.73%	3	1.63E+06	88.54	< 0.0005
Degree*Mobility	9.14E+05	0.70%	6	1.52E+05	8.28	< 0.0005
Shape*Degree*Mobility	1.01E+06	0.77%	6	1.68E+05	9.12	< 0.0005
Errors	9.27E+06	7.08%	504	1.84E+04		

$S_e = 135.65$ R^2-Adj = 92.6%

E-OP. Table 4.10 shows the ANOVA for E-OP's BC-Bits Received response. The first thing to note is that the p-value for the shape factor is 0.868. This indicates that there is an 86.8% probability that any variation explained by shape is actually due to random effects. Therefore, shape and its subsequent interactions are pooled from the ANOVA computations. The resulting ANOVA, Table 4.11, has a coefficient of determination of 92.79%. Mobility explains 86.49% of the variation in BC-Bits Received response, while degree level only explains 6.23% of the variation. Lastly, the interactions of degree and mobility only explains 0.18% of the variation, and the remaining 7.10% is experimental error.

4.5.6 Summary. The results of this analysis clarify what factors are the major contributors to variation in the overall communications overhead. The performance differences of the two algorithms is also evident.

Even though irregular network shape tends to have a negative effect on localization convergence and accuracy, it does not negatively impact the communications

Table 4.10: E-OP: Initial ANOVA for BC-Bits Received

Component	Sum of Squares	Percentage Variation	Degrees of Freedom	Mean Squares	F-Computed	P-Value
SSY	1.22E+09		720			
SS0	1.08E+09		1			
SST	1.38E+08		719			
Shape	287	0.00%	1	2.87E+02	0.03	0.868
Degree	8614549	6.23%	2	4.31E+06	415.20	< 0.001
Mobility	119695044	86.49%	3	3.99E+07	3846.01	< 0.001
Shape*Degree	842307	0.61%	2	4.21E+05	40.60	< 0.001
Shape*Mobility	1274271	0.92%	3	4.25E+05	40.94	< 0.001
Degree*Mobility	255698	0.18%	6	4.26E+04	4.11	< 0.001
Shape*Degree*Mobility	482832	0.35%	6	8.05E+04	7.76	< 0.001
Errors	7220276	5.22%	696	1.04E+04		
	$S_e = 101.85$		R_2-Adj = 94.61%			

Table 4.11: E-OP: ANOVA for BC-Bits Received

Component	Sum of Squares	Percentage Variation	Degrees of Freedom	Mean Squares	F-Computed	P-Value
SSY	1.22E+09		720			
SS0	1.08E+09		1			
SST	1.38E+08		719			
Degree	8.61E+06	6.23%	2	4.31E+06	310.55	< 0.0005
Mobility	1.20E+08	86.49%	3	3.99E+07	2876.59	< 0.0005
Degree*Mobility	2.56E+05	0.18%	6	4.26E+04	3.07	0.006
Error	9.82E+06	7.10%	708	1.39E+04		
	$S_e = 117.771$		R^2-Adj = 92.79%			

overhead required to perform localization. In fact, for both algorithms, the irregular C-shape networks tend to significantly lower the number of bits transmitted and received during localization. However, as expected, mobility and degree have a more significant effect. For the Map-Growing algorithm, the effects of degree and mobility are similar; as the degree and mobility levels increase, so does the communications cost of achieving localization.

It is interesting to see how the percent of variation explained by degree and mobility differ for the two algorithms. The multi-phase incremental approach used by Map-Growing consists of two flooding phases where the number of transmits and receives during those phases are highly dependant on the average degree of the network. This ultimately masks the amount of variation explained by mobility. Alternatively,

using APS-Euclidean's single phase concurrent approach, the effects of mobility overshadow the effects of degree.

Lastly, a comparison of the communications cost of the two algorithms shows that APS-Euclidean's mean responses for bits transmitted and received are significantly less than MG-OP at an $\alpha = 0.1$ significance level. Figure 4.16 further confirms these results.

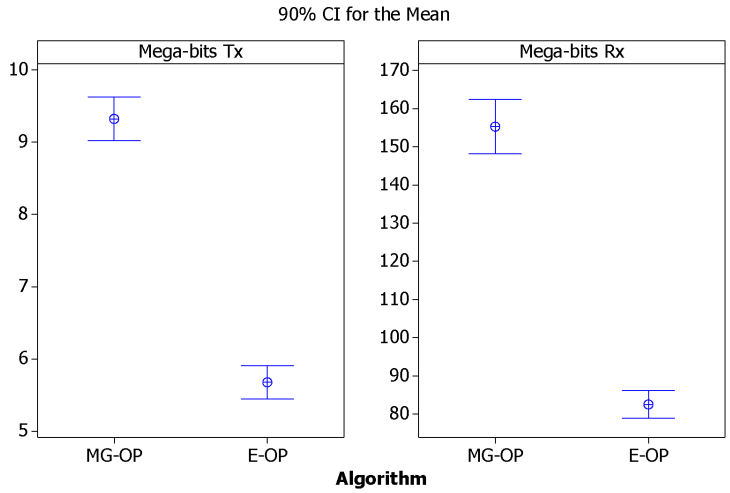

Figure 4.16: Interval Plots: Mean Mega-Bits Transmitted and Received Responses versus Algorithm

4.6 Power Models

This section presents power models useful for predicting the power consumption of a node during the localization process. Two models are derived. The first assumes optimal receiver capabilities, where a receiver is on only when a message is being received. Thus, the power consumed while receiving is based on the number of bits received. The second power model assumes the receiver is always on and therefore uses the time to localize to determine the power consumed while receiving. Both models assume receiving and processing are independent of one another. That is, a node's processor may be in sleep mode while the receiver is on, and vise versa. This section introduces the models, explains how they are derived, and presents example

predictions based on the algorithm and various levels of the factors varied. The general equation for power consumed is

$$C_p = [a_0 T + a_1 R + a_2 P]\text{mA-hours} \tag{4.1}$$

where a_0 and a_1 are the predicted Bits Transmitted and Received responses, and a_2 is an estimate of the worst case number of instructions required for a single node to localize. The values for a_0 and a_1 are obtained via the reverse transformation of the multiple linear regression equations for BC-Bits Transmitted and BC-Bits Received, respectively. Parameters T, R, and P represent the total current used in mA-hours for transmitting a_0 bits, receiving a_1 bits, and processing a_2 instructions respectively. Suppose transmissions use X mA, receives use Y mA, processing uses Z mA, a processor capable of executing i instructions per second, and a transmit and receive data rate of j bits per second, the equations in Table 4.12 define parameters T, R, and P. The 3,600 in the denominator of T and P converts the transmit and receive data rate from bits/second to bits/hour.

Table 4.12: Parameter Summary for Power Models

Algorithm / Parameter	Parameter Value
$T =$	$\frac{X}{3600 \times j}$ mA-hours/bit
$R =$	$\frac{Y}{3600 \times j}$ mA-hours/bit
$P =$	$\frac{Z}{3600 \times i}$ mA-hours/instruction

Section 4.6.1 derives parameters a_0 and a_1, while Section 4.6.2 discusses the derivation of the estimate for a_3. Finally, Section 4.6.3 contains an example power predication and analysis of the results.

4.6.1 Multiple Linear Regression. This section presents the results of multiple linear regression for the BC-Bits Transmitted and BC-Bits Received responses, derived using MiniTab 14.

Earlier in Section 4.5, the visual tests for independent and normally distributed residuals with a constant standard deviation are discussed and verified. However, a linear relationship between predictors and the response is also necessary to perform multiple linear regression. The scatter plots of BC-Bits Received versus degree and mobility, and BC-Bits Transmitted versus degree and mobility for both algorithms are provided in Appendix C. These plots verify that the relationship between the factors and the responses are in fact linear.

Tables 4.13, 4.14, 4.15, and 4.16 summarize the results of the multiple linear regression for BC-Bits Transmitted and Received for both algorithms. The b_i columns are the coefficient of the predictor and is used directly in the regression equations shown later in Table 4.17. The s_{b_i} column is the standard error of the coefficients and is used to calculate the confidence intervals. Given the degrees of freedom of the error term is 524 for MG-OP and 715 for E-OP, the 90% confidence intervals of the predictor terms are calculated using the z-variate, $z_{\alpha=0.1} = 1.645$. Additionally, all p-values are less than 0.0005, indicating that each of the predictors are significant. Predictors are selected based on their p-value and their impact on the coefficient of determination. However, to simplify the regression, higher-order predictors are only used if the sum of the increase in R^2-Adj is greater than 1%.

Table 4.13: MG-OP: Regression Parameters for BC-Bits Transmitted

Predictor	b_i	s_{b_i}	p-value	90% Confidence Interval
Constant	287.46	9.30	<0.0005	(272.16, 302.75)
Shape	-41.51	2.26	<0.0005	(-45.23, -37.78)
Degree	23.23	0.69	<0.0005	(22.09, 24.37)
Mobility	319.77	11.79	<0.0005	(300.38, 339.16)
	$S_e = 52.0187$		R^2-Adj = 80.6%	

Table 4.14: MG-OP: Regression Parameters for BC-Bits Received

Predictor	b_i	S_{b_i}	p-value	90% Confidence Interval
Constant	441.48	36.58	<0.0005	(381.31, 501.65)
Shape	-136.60	8.91	<0.0005	(-151.25, -121.95)
Degree	93.89	2.73	<0.0005	(89.41, 98.38)
Mobility	1595.64	46.38	<0.0005	(1519.34, 1671.94)
$S_e = 204.652$			R^2-Adj = 83.20%	

Table 4.15: E-OP: Regression Parameters for BC-Bits Transmitted

Predictor	b_i	S_{b_i}	p-value	90% Confidence Interval
Constant	275.91	6.68	<0.0005	(264.92, 286.90)
Shape	-21.97	2.63	<0.0005	(-26.30, -17.65)
Degree	3.68	0.40	<0.0005	(3.02, 4.35)
Mobility	1232.49	26.38	<0.0005	(1189.09, 1275.89)
$Mobility^2$	-1051.72	50.60	<0.0005	(-1134.96, -968.48)
$S_e = 35.2672$			R^2-Adj = 93.9%	

Table 4.16: E-OP: Regression Parameters for BC-Bits Received

Predictor	b_i	S_{b_i}	p-value	90% Confidence Interval
Constant	431.51	18.49	<0.0005	(401.09, 461.93)
Degree	33.37	1.38	<0.0005	(31.10, 35.64)
Mobility	3376.49	90.36	<0.0005	(3227.85, 3525.13)
$Mobility^2$	-2551.20	173.40	<0.0005	(-2836.44, -2265.96)
$S_e = 25.58$			R^2-Adj = 92.4%	

Given these regression parameters, the generic regression equations take the form of

$$y_p = b_0 + b_1 x_{1_p} + b_2 x_{2_p} + \cdots + b_k x_{k_p} \tag{4.2}$$

where y_p is the predicted response and the x_{i_p} values are the input levels that correspond to the predictor coefficients, b_i.

The R^2-Adj values indicate between 80.6% and 93.9% of the variation in the responses is explained by these regressions. Although the lower R^2-Adj values indicate less accurate predictions of the response, they are still useful for determining a rough-order-of-magnitude estimate fits the measured response.

4.6.1.1 Prediction Interval. The regression equations predict or estimate the response of the algorithms for various factor levels. The 90% confidence intervals associated with the predicted results are derived using the standard deviation of the predictions

$$S_{\hat{y}_p} = S_e \sqrt{\frac{1}{m} + x_p{}^T C x_p} \tag{4.3}$$

where s_e is the standard deviation of error values (given in the Regression Parameter Tables), x_p is the single column matrix of predictor levels, and C is the correlation matrix of regression variables (provided in Appendix D) [Jai91]. Using the z-variate 1.645, it follows the 90% confidence intervals for the predicted results are

$$y_p \mp 1.645 s_{\hat{y}_p}. \tag{4.4}$$

The matrix x_p contains the input levels of the predictors. For example, using the APS-Euclidean BC-Bits Transmitted response, to predict the response for a square shape network with a degree level of 14 and a mobility level of 40%, the x_p is

$$x_p = \begin{bmatrix} 1 \\ 0 \\ 14 \\ 0.4 \\ 0.16 \end{bmatrix}. \tag{4.5}$$

4.6.2 Processing Time. Without constructing a simulation of a processor, or implementing the localization algorithms on hardware, there is no practical way to precisely determine the processing time of localization. Therefore, a worst case estimate is calculated for use in the power models. The OPNET custom process model C-code is disassembled to Intel x8086 assembly instructions. Using the highest observed node degree and the highest number of received messages by a single node as

inputs, the assembly code is examined, and a upper-limit on the number of instructions executed by a single node is determined for both algorithms. The estimated worst case number of instructions executed for a single node using MG-OP is $a_2 = 150,000,000$, and for E-OP $a_2 = 2,400,000,000$ instructions. E-OP requires sixteen times more processing than MG-OP, because the method used to obtain *anchor* distances are much more computationally intensive than the methods used for localizing in MG-OP. Also, once a node localizes in MG-OP it has no significant computations left to perform with the exception of the affine transformation. Alternatively, E-OP requires a localized node to continue to compute distance estimates to additional *anchors*, to aide and improve the localization of other nodes. Even though MG-OP generate more messages than E-OP on average, the E-OP simulations were still somewhere about twenty times slower than MG-OP simulations. This intuitively indicates that the ratio of the two estimates are proportionally accurate.

Table 4.17 summarizes the equations for a_0 and a_1, and the worst case a_2 estimates for both MG-OP and E-OP.

Table 4.17: Parameter Summary for Power Models

Algorithm/ Parameter	Parameter Value
MG-OP	
$a_0 =$	$\frac{(287.46-41.51\times Shape+23.23\times Degree+319.77\times Mobility)^{2.5}}{200}$ bits
$a_1 =$	$\frac{(441.48-136.60\times Shape+93.89\times Degree+1595.64\times Mobility)^{2.5}}{200}$ bits
$a_2 =$	150,000,000 Instructions
E-OP	
$a_0 =$	$\frac{(275.91-21.97\times Shape+3.68\times Degree+1232.49\times Mobility-1051.72\times Mobility^2)^{2.5}}{200}$ bits
$a_1 =$	$\frac{(431.51-33.37\times Degree+3376.49\times Mobility-2251.20\times Mobility^2)^{2.5}}{200}$ bits
$a_2 =$	2,400,000,000 Instructions

4.6.3 Power Model Results. This section uses the power models discussed above to estimate the power required to localize using both algorithms. An example estimate and an analysis of the results are discussed. The estimates are based on a full-square network with degree 14 and a mobility level of 40%. Thus, the input matrix, x_p, is the same as shown in (4.5).

4.6.3.1 Power Models with Optimal Receiver. Using x_p as input to the regression equations given in Table 4.17, the parameters for a_0 and a_1 are calculated and shown in Table 4.18. This table also includes the 90% confidence intervals for the predicted mean response that correspond to having m equal to 1, 30, or 100 future observations.

Table 4.18: Example Predictions for Mega-Bits Transmitted and Received

Algorithm	Parameter	Mean	90%CI (m = 1)	90%CI (m = 30)	90%CI (m = 100)
MG-OP	a_0	14.92	(10.97, 19.63)	(14.08, 15.80)	(14.38, 15.48)
	a_1	280.69	(191.80, 390.09)	(261.32, 300.48)	(268.15, 293.16)
E-OP	a_0	10.86	(8.56, 13.48)	(10.32, 11.40)	(10.48, 11.26)
	a_1	127.91	(94.43, 167.48)	(120.70, 135.39)	(123.17, 132.78)

Table 4.19 provides example values for the variables used in defining parameters T, R, and P. These values correspond to the specifications of Crossbow Technologies MICA2DOT Wireless Microsensor Mote [Xbo05]. It is assumed for these calculations, that the mote's 4MHz processor is capable of executing one instruction per clock cycle. Given these values for a_i, T, R, and P, the predictions are computed and shown in Table 4.20. The results highlight two interesting findings. First of all, the power consumed by a node due to the data communications overhead is extremely small, even with node mobility. In fact, even though the MG-OP algorithm required twice the amount of power for data communications (0.116097 mA-hr) compared to APS-Euclidean (0.056873 mA-hr), it still only requires 0.04% of the battery capacity. Interestingly though, the estimated power required for processing given the APS-Euclidean algorithm is substantially more than the power costs associated with

Table 4.19: Power Model Example Parameter Values

Algorithm / Parameter	Parameter Value
$T =$	$\frac{27}{3{,}600 \times 38{,}400}$ mA-hr/bit
$R =$	$\frac{10}{3{,}600 \times 38{,}400}$ mA-hr/bit
$P =$	$\frac{8}{3{,}600 \times 4{,}000{,}000}$ mA-hr/instruction

transmitting and receiving data. Consider, however, that this estimate is based on rough-order-of-magnitude worst case scenario, with no computational optimizations assumed. Given a receiver with optimal capabilities, these results show that the power associated with data communications during node localization is extremely small.

Table 4.20: Power Model Example Predictions

Algorithm	a_0T (mA-hr)	a_1R (mA-hr)	a_2P (mA-hr)	Total Power Consumed (mA-hr)	% of Battery (560mA-hr Capacity)
MG-OP	0.01	0.10	0.08	0.20	0.04%
E-OP	0.01	0.05	1.11	1.17	0.21%

4.6.3.2 Power Models with Receiver-Always-On. To ensure nodes receive all incoming messages, a synchronized Medium Access Control (MAC) protocol could be used, or alternatively the receivers could be on at all times. Since these experiments do not model a MAC protocol, the power model assumes that receivers are always on during the localization process. Thus estimating the worst case power required to receive.

This assumption means the total power required for receiving, a_1R, is calculated based on the time it takes to localize instead of the number of bits received. The total power required for transmitting, a_0T, and the total power required for processing, a_2P, derived above remain unchanged.

Consequently, the experimental results are examined to estimate the total time for the localization algorithms to converge. First, it is assumed the time to localize is dependent on the mobility level of the network. The following estimates are based on the average time to localize and the general trends in the response. The static network localizes in the shortest amount of time (10 minutes), while the networks with the highest level of mobility take the longest amount of time to localize (40 minutes for MG-OP and 25 minutes for E-OP). The expressions for a_1 in Table 4.21 are used to estimate the time to localize for both algorithms. For these power models, the receive power parameter, R, is given by the hardware specifications of the sensor platform.

Table 4.21: Alternate Receive Power Parameters

Algorithm	a_1
MG-OP	$[\frac{1}{6} + Mobility]$ hours
E-OP	$[\frac{1}{6} + (0.5 \times Mobility)]$ hours

Subsequently, a_1 and R are multiplied to determine the power required to receive for the entire a_1 hours. Table 4.22 shows the updated estimate for the power consumed during localization, given the same factor levels used in the earlier estimate. Having the receiver always on in this second scenario significantly increases the overall power required to localize for both algorithms. However, the power consumed by a node is still relatively small, even with node mobility. For instance, even a network with a 60% mobility level and a degree of 18 consumes less than 2% of the 560mA-hr battery.

Thus, localization algorithms similar to Map-Growing and APS-Euclidean should concentrate on achieving position accuracy, scalability, and the ability to converge, rather than the power costs associated with data communications overhead.

Table 4.22: Power Model Example Predictions

Algorithm	a_0T (mA-hr)	a_1R (mA-hr)	a_2P (mA-hr)	Total Power Consumed (mA-hr)	% of Battery (560mA-hr Capacity)
MG-OP	0.01	5.67	0.08	5.76	1.03%
E-OP	0.01	3.67	1.11	4.79	0.86%

4.7 Size-Factor Analysis

The number of nodes in a network has a significant affect on the mean position error performance of the Map-Growing algorithm. Therefore, a secondary experiment using the number of nodes in the network as a factor is performed to compare the scalability of the two algorithms. Networks consisting of 100, 200, 400, and 800 nodes are tested with varying average node degree of degree 12 and 16. The results of these experiments, shown in Figure 4.17, indicate that the position error performance of the APS-Euclidean algorithm is essentially independent of the network size, while Map-Growing is directly affected by the size of the network. That is, as network size increases, so does the resulting mean position error.

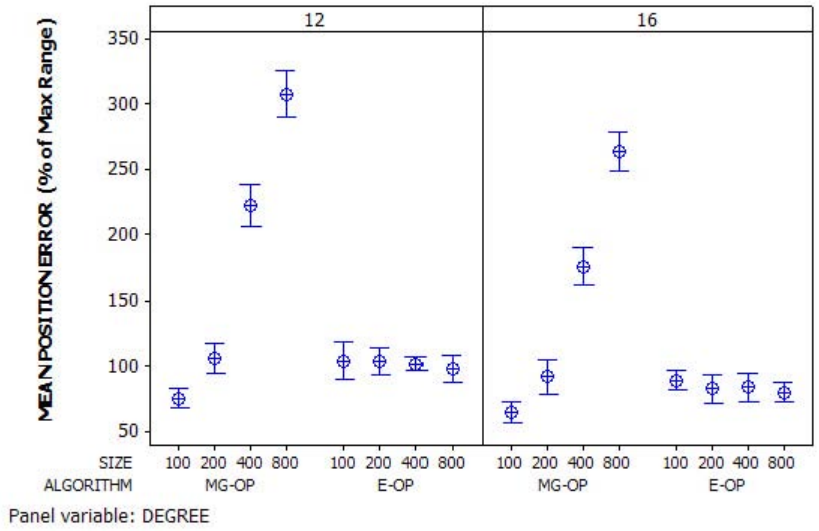

Figure 4.17: Graph of Network Size Effects

Using visual tests on the 90% confidence intervals, MG-OP achieves better mean position error performance for network sizes of 100, at a significance level of $\alpha = 0.1$. The mean position error performance of the two algorithms is not significantly different for network sizes of 200. However, MG-OP achieves a significantly worse mean position error for networks of size 400 and 800.

Since Map-Growing is a incremental algorithm, measurement errors originate at a single starting point and continue to propagate and compound as the network grows. The incremental approach used in the Map-growing algorithm is therefore not suitable for larger networks with moderate ranging error. However, an incremental approach such as Map-Growing may be useful in networks with smaller range error, networks with as few as 3 *anchor* nodes, and networks with a small number of nodes ($n \leq 200$). These results emphasize the inherent strength in APS-Euclidean (a concurrent non-shortest path algorithm), lies in its scalable and predictable performance for larger networks.

4.8 Summary

This chapter discusses the effects of network shape, network degree, and node mobility on the communications overhead associated with performing localization in an AWSN. The experiments show that mobility and degree account for the majority of variation in the Bits Transmitted and Bits Received responses, while network shape has relatively little effect. It is also determined that the concurrent localization approach used by the APS-Euclidean algorithm results in better localization performance with a lower communications overhead than the incremental approach used by Map-Growing. Power models are developed to predict the power costs of performing localization in a 200 node mobile AWSN. Lastly, power costs relating to data communications are shown to be small with respect to the capacity of an average battery.

V. Conclusions

5.1 Introduction

This chapter reviews the objectives and the respective experiments of this research. Additionally, a summary of the research conclusions and contributions are provided. Last, potential areas of future research are discussed.

5.2 Conclusions of Research

This research examined the effects of network shape, average node degree, and node mobility on the performance of AWSN localization. The research first analyzed the effects of these factors on position estimation accuracy and the localization algorithms to converge. However, the main goal of the research was to determine the effect of shape, degree, and mobility on data communications overhead and the corresponding power costs required to localize. Additionally, the Map-Growing and the APS-Euclidean algorithms were compared.

A few interesting conclusions relating to mean position error and mean percent localized responses come from this research. For one, the resulting metrics show that localization algorithms have a more difficult time localizing in irregular shaped networks and low degree networks. Specifically, the incremental Map-Growing algorithm has the most difficulty, only achieving 75% localization in C-Shaped networks, and 68% in degree 8 networks. This finding indicates the incremental approach used in Map-Growing is not suitable for low degree and irregularly-shaped networks. Interestingly, it is also determined that a low percentage of node mobility, 10% to 30%, can significantly improve the convergence of a localization algorithm. Low mobility levels effectively allow disconnected or poorly positioned nodes to move to a better position, while not negatively impacting the ability of localization to converge. However, 50% node mobility negates this effect, and ultimately causes an incremental localization method such as Map-Growing to perform both poorly and unpredictably.

The conclusion reached based on the first objective of this research is that mobility and node degree account for the majority of variation in the bits transmitted

and bits received. For both algorithms, the effects of degree and mobility are similar; as the degree and mobility levels increase, so does the communications cost of achieving localization. For Map-Growing, the percentage of response variation explained by the two factors are comparable. However, for APS-Euclidean, mobility accounts for 92.91% of the variation in Bits Transmitted and 86.49% in Bits Received. Additionally, although an irregular network shape tends to have a negative effect on localization convergence and accuracy, it does not negatively impact the communications overhead required to perform localization.

The second objective, to compare the performance of an incremental versus a concurrent localization algorithm, found that the concurrent localization approach used by the APS-Euclidean algorithm results in better overall localization performance while requiring significantly less communications overhead than the incremental approach used by Map-Growing. Specifically, APS-Euclidean localized a significantly higher percentage of nodes with a significantly lower mean position error than the Map-Growing algorithm. APS-Euclidean is also more predictable, given that it performs better with node mobility and it is more scalable. It is evident that because the APS-Euclidean algorithm requires less communications overhead and because it achieves better accuracy and convergence, it is the more stable and predictable algorithm of the two. However, this is at the expense of processing time, given an estimated 20 times more processing required compared to Map-Growing.

This research also develops Power Models to predict the power costs of performing localization in a 200 node mobile AWSN. Estimates are presented and analyzed. Assuming the receiver is only on when receiving messages shows the power associated with the data communications overhead of the localization algorithms is less than 0.02% of an average button-cell battery with a 560mA-hr capacity. Whereas, assuming the worst case scenario where a node's receiver must remain on to ensure all messages are received, the results show that even in a high mobility high degree configuration, less than 2% of the battery is consumed. Regardless, the total power cost associated with data transmissions and processing time is relatively small.

Given these results, it is apparent that the power associated with communications and processing is not the most critical aspect or concern of localization in AWSNs. On the contrary, the primary focus in the development of localization algorithms should be position estimation accuracy, robustness, and scalability.

5.3 Research Contributions

The following are the main contributions of this research:

- Network shape does not negatively impact data communications overhead in node localization using APS-Euclidean and Map-Growing like algorithms.
- The average network degree and mobile nodes have the most significant effect on the data communications overhead in node localization. As both factors increase, so does the data communications overhead.
- Incremental localization approaches such as Map-Growing are not suitable for use in networks with low degree, high percentage of node mobility, or networks with more than 200 nodes.
- Even with high mobility and high degree networks, the overall data communications overhead associated with localization only consumes a small percentage of a battery capacity.

5.4 Recommendations for Future Work

There are many additional areas of research in node localization for wireless sensor networks. Specifically, follow-on research in these areas could be examined:

- **Self-Organizing Mobile Networks:** Given the effect of low levels of mobility, it may be useful to examine 'smart' mobility algorithms to allow self-configuring optimal placement and distribution of nodes. For example, targeting an optimal average node degree and a node distribution would likely show significant improvement in the accuracy, convergence, and possibly the size of the area 'covered' by the sensor network.

- **Mobile *Anchors*:** Along the same line as self-organizing mobile networks, it would be interesting to examine the possible modification of specialized *anchor* placement algorithms such as HEAP and STROBE [BHE01], for self-organizing mobile *anchors*. The goal of this work would also be to improve position estimation accuracy, localization convergence, and area coverage of a sensor network.
- **Examine Processing Costs:** Lastly, it would be interesting to examine the computational complexity and associated processing requirements of localization algorithms. Specifically, given the processing estimates of this research, examining the real-time processing requirements for APS-Euclidean and/or other localization algorithms would be useful in determining the significance of power costs associated with processing required for localization. Given a high computational complexity, further examining tradeoffs of computational optimizations verses the accuracy and convergence of the algorithms would be fruitful.

5.5 Summary

This chapter reviews the objectives and the respective experiments of this research. The effects of node mobility, network degree, and network shape on node localization performance is discussed. This research concludes that node mobility and network degree have the largest affect on the bits transmitted and received during localization, and therefore account for the largest percentages of variation in the responses. Lastly, the power cost associated with the data communication overhead is relatively small percentage of an average battery capacity.

Appendix A. Position Estimation Techniques

A.1 Trilateteration

Trilateration is a form of triangulation, in which an *unknown* node uses the estimated distances, d_i, to known positions (x_i, y_i) of three *anchors* in order to node to estimate it position (x, y). Trilateration begins with deriving the system of equations of the three circles centered around each of the *anchor* nodes with respective radii, d_i [LR03]. The system of equations shown in (A.1), (A.2), and (A.3).

$$(x_1 - x)^2 + (y_1 - y)^2 = {d_1}^2 \tag{A.1}$$

$$(x_2 - x)^2 + (y_2 - y)^2 = {d_2}^2 \tag{A.2}$$

$$(x_3 - x)^2 + (y_3 - y)^2 = {d_3}^2 \tag{A.3}$$

The system is linearized by subtracting (A.3) from (A.1) and (A.2). The respective results are shown in (A.4) and (A.5).

$${x_1}^2 - {x_3}^2 - 2(x_1 - x_3)x + {y_1}^2 - {y_3}^2 - 2(y_1 - y_3)y = {d_1}^2 - {d_3}^2 \tag{A.4}$$

$${x_2}^2 - {x_3}^2 - 2(x_2 - x_3)x + {y_2}^2 - {y_3}^2 - 2(y_2 - y_3)y = {d_2}^2 - {d_3}^2 \tag{A.5}$$

The linear equations are then placed in the form $Ax = b$, as shown in (A.6).

$$\begin{bmatrix} 2(x_1 - x_3) & 2(y_1 - y_3) \\ 2(x_2 - x_3) & 2(y_2 - y_3) \end{bmatrix} x = \begin{bmatrix} {x_1}^2 - {x_3}^2 + {y_1}^2 - {y_3}^2 + {d_3}^2 - {d_1}^2 \\ {x_2}^2 - {x_3}^2 + {y_2}^2 - {y_3}^2 + {d_3}^2 - {d_2}^2 \end{bmatrix} \tag{A.6}$$

This system is then solved using the standard least-squares approach shown in (A.7).

$$\begin{bmatrix} x & y \end{bmatrix} = (A^T A)^{-1} A^T b \tag{A.7}$$

A.1.1 Residue. The residue of the position estimate (x, y) is computed as shown in (A.8) to determine the consistency of the set of equations [LR03].

$$Residue = \frac{\sum_{i-1}^{n} \sqrt{(x_1 - x)^2 + (y_1 - y)^2} - d_i}{3} \tag{A.8}$$

A residue of zero indicates a position estimate that is exact given the estimated distances. As the residue increases so does the inconsistency and/or inaccuracy of the position estimate.

A.2 2-Anchor Localize Method

The 2-*Anchor* Localize method consists of using the estimated distances, d_i, to known positions (x_i, y_i) of two *anchors*, to determine the two positions where an *unknown* node may exist.

The method begins by deriving (A.9) and (A.10), which are the equations of the circles based on the estimated distances, d_i, and positions (x_i,y_i) to the two known *anchor* nodes.

$$(x_p - x_1)^2 + (y_p - y_1)^2 = d_1^2 \tag{A.9}$$

$$(x_p - x_2)^2 + (y_p - y_2)^2 = d_2^2 \tag{A.10}$$

Subsequently, (A.12), the equation of the straight line intersecting the the two points where the circles intersect, is derived by setting (A.9) equal to (A.10) and solving for y.

$$y_p = mx_p + c \tag{A.11}$$

$$y_p = \frac{(-2x_1 + 2x_2)}{(2y_1 - 2y_2)} x_p + \frac{(x_1^2 - x_2^2 + y_1^2 - y_2^2 - d_1^2 + d_2^2)}{(2y_1 - 2y_2)} \tag{A.12}$$

Next, (A.12) is substituted into either one of the two circle equations, A.9 or A.10, and rearranged into quadratic form, as shown in (A.13).

$$(1+m^2)x_p^2 + (2m(c-y_i) - 2x_i)x_p + x_i^2 + (c-y_i)^2 - d_i^2 = 0 \tag{A.13}$$

It follows that the quadratic formula is used to solve (A.13), yielding the two x-coordinates shown in (A.14) and (A.15).

$$x_{p1} = -b + \frac{\sqrt{a^2+b^2-2ac}}{2a} \tag{A.14}$$

$$x_{p2} = -b - \frac{\sqrt{a^2+b^2-2ac}}{2a} \tag{A.15}$$

The corresponding y_p coordinates are solved for by substituting x_{p1} and x_{p2} into (A.11), and solving for y respectively.

A.3 Affine Coordinate Transformation

Given three "control points" (i.e., *Anchor* nodes), whose coordinates are known in two different coordinate systems (relative and global), the relative coordinates of an *unknown* can be transformed into the corresponding global coordinate system, using Affine Coordinate Transformation [WG97]. The affine transformation applies an x and y scale factor, two translations of the origin, a rotation about the origin, and a "small nonorthogonality correction between the x and y axes" [WG97]. These six unknowns form the following mathematical model for affine transformation

$$X = ax + by + c \tag{A.16}$$

$$Y = dx + ey + f \tag{A.17}$$

Given the respective relative and global (x,y) coordinates of three *anchors* (a_1,a_2,a_3), the system of equations shown in (A.18) is formed. Similar to Latera-

tion, the equations are placed in the form, Ax = L, and solved using the least squares Equation shown in (A.19).

$$\begin{bmatrix} x_{a_1} & y_{a_1} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{a_1} & y_{a_1} & 1 \\ x_{a_2} & y_{a_2} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{a_2} & y_{a_2} & 1 \\ x_{a_3} & y_{a_3} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{a_3} & y_{a_3} & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} X_{a_1} \\ Y_{a_1} \\ X_{a_2} \\ Y_{a_2} \\ X_{a_3} \\ Y_{a_3} \end{bmatrix} \tag{A.18}$$

$$x = (A^T A)^{-1} A^T L \tag{A.19}$$

Appendix B. Experimental Data Analysis Tables

Table B.1: MG-OP: Computation of Effects for Mean Percent Localized Response

		Mobility (%)								
Shape	*Degree Level*	0	10	30	50	Row Sum	Row Mean	Row Effect	**Shape Effect**	**Degree Effect**
Square	Low	73.82	95.80	94.57	84.93	7680.50	87.28	1.97	9.58	-17.18
	Medium	97.77	99.82	99.11	95.70	8633.00	98.10	12.79		5.30
	High	98.84	99.93	99.68	98.75	8738.50	99.30	13.99		11.89
C-Shape	Low	35.07	67.14	55.34	38.36	4310.00	48.98	-36.34	-9.58	
	Medium	85.91	99.23	81.95	65.39	7314.50	83.12	-2.19		
	High	98.77	99.45	99.11	83.05	8368.50	95.10	9.78		
	Col Sum	10784.00	12350.00	11655.00	10256.00	45045.00				
	Col Mean	81.70	93.56	88.30	77.70		**85.31**			
	Mobility Effect	-3.62	8.25	2.98	-7.62					

Table B.2: MG-OP: 90% Confidence Intervals for Main Effects on Mean Percent Localized Response(% of Max Range)

Parameter	Mean Effect	Std Dev	90% Confidence Interval
Mean	85.31	0.35	(84.74, 85.88)
Mobility Effects			
Static	-3.62	0.60	(-4.60, -2.63)
10% Mobile	8.25	0.60	(7.26, 9.23)
30% Mobile	2.98	0.60	(2.00, 3.97)
50% Mobile	-7.62	0.60	(-8.60, -6.63)
Degree Effects			
Low	-17.18	0.49	(-17.99, -16.38)
Med	5.30	0.49	(4.49, 6.10)
High	11.89	0.49	(11.08, 12.69)
Shape Effects			
Square	9.58	0.35	(9.01, 10.15)
C-Shaped	-9.58	0.35	(-10.15, -9.01)

Table B.3: E-OP: Computation of Effects for Mean Percent Localized Response (% of Max Range)

		Mobility (%)								
Shape	*Degree Level*	0	10	30	50	Row Sum	Row Mean	Row Effect	**Shape Effect**	**Degree Effect**
Square	Low	86.90	95.53	92.28	87.63	10870.50	90.59	-1.90	2.39	-8.38
	Medium	97.53	96.75	95.42	95.10	11544.00	96.20	3.71		2.88
	High	99.65	97.63	97.10	97.05	11743.00	97.86	5.37		5.50
C-Shape	Low	64.40	86.83	83.60	75.72	9316.50	77.64	-14.85	-2.39	
	Medium	90.20	96.62	96.67	94.68	11345.00	94.54	2.05		
	High	98.42	98.05	98.08	97.93	11774.50	98.12	5.63		
	Col Sum	16113.00	17142.50	16894.50	16443.50	66593.50				
	Col Mean	89.52	95.24	93.86	91.35		**92.49**			
	Mobility Effect	-2.97	2.75	1.37	-1.14					

Table B.4: E-OP: 90% Confidence Intervals for Main Effects on Mean Percent Localized Response(% of Max Range)

Parameter	Mean Effect	Std Dev	90% Confidence Interval
Mean	92.49	0.18	(92.19, 92.79)
Mobility Effects			
Static	-2.97	0.32	(-3.50, -2.45)
10% Mobile	2.75	0.32	(2.22, 3.27)
30% Mobile	1.37	0.32	(0.85, 1.89)
50% Mobile	-1.14	0.32	(-1.66, -0.62)
Degree Effects			
Low	-8.38	0.26	(-8.80, -7.95)
Med	2.88	0.26	(2.45, 3.31)
High	5.50	0.26	(5.07, 5.92)
Shape Effects			
Square	2.39	0.18	(2.09, 2.69)
C-Shaped	-2.39	0.18	(-2.69, -2.09)

Table B.5: MG-OP: Computation of Effects for Mean Position Error Response (% of Max Range)

		Mobility (%)								
Shape	*Degree Level*	0	10	30	50	Row Sum	Row Mean	Row Effect	**Shape Effect**	**Degree Effect**
Square	Low	128.17	158.45	111.01	72.75	10348.38	117.60	8.67	1.97	-11.90
	Medium	128.11	123.24	117.77	90.38	10109.02	114.88	5.95		13.24
	High	100.50	109.37	103.06	87.92	8818.57	100.21	-8.72		-1.34
C-Shape	Low	67.86	120.97	69.08	47.91	6728.17	76.46	-32.47	-1.97	
	Medium	156.50	160.54	111.53	89.25	11391.98	129.45	20.53		
	High	121.62	134.23	122.02	82.03	10117.67	114.97	6.05		
	Col Sum	15460.72	17749.52	13958.39	10345.16	57513.78				
	Col Mean	117.13	134.47	105.75	78.37		**108.93**			
	Mobility Effect	8.20	25.54	-3.18	-30.56					

Table B.6: MG-OP: 90% Confidence Intervals for Main Effects on Mean Position Error Response (% of Max Range)

Parameter	Mean Effect	Std Dev	90% Confidence Interval
Mean	108.93	1.83	(105.92, 111.94)
Mobility Effects			
Static	8.20	3.17	(2.98, 13.42)
10% Mobile	25.54	3.17	(20.32, 30.75)
30% Mobile	-3.18	3.17	(-8.40, 2.03)
50% Mobile	-30.56	3.17	(-35.77, -25.34)
Degree Effects			
Low	-11.90	2.59	(-16.16, -7.64)
Med	13.24	2.59	(8.98, 17.50)
High	-1.34	2.59	(-5.59, 2.92)
Shape Effects			
Square	1.97	1.83	(-1.05, 4.98)
C-Shaped	-1.97	1.83	(-4.98, 1.05)

Table B.7: E-OP: Computation of Effects for Mean Position Error Response (% of Max Range)

		Mobility (%)								
Shape	*Degree Level*	0	10	30	50	Row Sum	Row Mean	Row Effect	**Shape Effect**	**Degree Effect**
Square	Low	129.30	121.94	108.35	88.30	13436.63	111.97	16.20	-1.48	10.68
	Medium	96.30	93.88	88.77	89.12	11042.06	92.02	-3.75		1.82
	High	81.85	79.88	77.38	76.39	9465.13	78.88	-16.89		-12.50
C-Shape	Low	98.17	122.30	98.60	84.61	12110.40	100.92	5.15	1.48	
	Medium	101.17	116.08	102.12	93.32	12380.79	103.17	7.40		
	High	91.70	89.75	87.16	82.05	10519.66	87.66	-8.11		
	Col Sum	17954.96	18714.87	16871.37	15413.48	68954.67				
	Col Mean	99.75	103.97	93.73	85.63		**95.77**			
	Mobility Effect	3.98	8.20	-2.04	-10.14					

Table B.8: E-OP: 90% Confidence Intervals for Main Effects on Mean Position Error Response (% of Max Range)

Parameter	Mean Effect	Std Dev	90% Confidence Interval
Mean	95.77	0.88	(94.33, 97.21)
Mobility Effects			
Static	3.98	1.52	(1.48, 6.48)
10% Mobile	8.20	1.52	(5.70, 10.70)
30% Mobile	-2.04	1.52	(-4.54, 0.46)
50% Mobile	-10.14	1.52	(-12.64, -7.64)
Degree Effects			
Low	10.68	1.24	(8.63, 12.72)
Med	1.82	1.24	(-0.22, 3.87)
High	-12.50	1.24	(-14.54, -10.46)
Shape Effects			
Square	-1.48	0.88	(-2.93, -0.04
C-Shaped	1.48	0.88	(0.04, 2.93)

Table B.9: MG-OP: Computation of Effects for Mega-bits Transmitted

		Mobility (%)								
Shape	*Degree Level*	0	10	30	50	Row Sum	Row Mean	Row Effect	**Shape Effect**	**Degree Effect**
Square	Low	4.51	6.48	9.13	13.32	735.74	8.36	-0.96	1.43	-3.28
	Medium	7.36	8.55	11.38	16.94	973.21	11.06	1.74		0.25
	High	9.25	10.45	13.67	18.05	1131.06	12.85	3.53		3.03
C-Shape	Low	1.97	4.04	4.28	4.62	327.95	3.73	-5.60	-1.43	
	Medium	6.24	8.14	8.36	9.61	711.70	8.09	-1.24		
	High	9.28	10.35	13.29	14.52	1043.41	11.86	2.53		
	Col Sum	849.20	1056.30	1322.25	1695.32	4923.08				
	Col Mean	6.43	8.00	10.02	12.84		**9.32**			
	Mobility Effect	-2.89	-1.32	0.69	3.52					

Table B.10: E-OP: Computation of Effects for Mega-bits Transmitted

Shape	*Degree Level*	*Mobility (%)* 0	10	30	50	Row Sum	Row Mean	Row Effect	**Shape Effect**	**Degree Effect**
Square	Low	1.50	3.51	7.05	10.79	685.09	5.71	0.03	0.39	-0.51
	Medium	1.44	3.20	7.80	11.25	710.74	5.92	0.25		0.14
	High	1.29	3.33	8.66	13.00	788.47	6.57	0.89		0.36
C-Shape	Low	0.74	3.44	6.04	8.31	555.80	4.63	-1.05	-0.39	
	Medium	1.64	3.87	7.19	10.17	686.04	5.72	0.04		
	High	1.62	3.42	7.08	9.94	661.54	5.51	-0.16		
	Col Sum	246.76	623.27	1313.96	1903.69	4087.68				
	Col Mean	1.37	3.46	7.30	10.58		**5.68**			
	Mobility Effect	-4.31	-2.21	1.62	4.90					

Table B.11: MG-OP: Computation of Effects for Mega-bits Received

Shape	*Degree Level*	*Mobility (%)* 0	10	30	50	Row Sum	Row Mean	Row Effect	**Shape Effect**	**Degree Effect**
Square	Low	41.38	73.70	139.40	272.60	11595.81	131.77	-23.51	26.10	-65.25
	Medium	90.15	114.40	175.53	350.40	16070.36	182.62	27.34		-2.43
	High	145.26	169.95	242.31	361.53	20219.03	229.76	74.48		67.68
C-Shape	Low	16.96	44.58	58.55	73.10	4250.08	48.30	-106.98	-26.10	
	Medium	74.51	106.20	131.99	179.60	10830.76	123.08	-32.20		
	High	147.99	171.49	241.28	303.89	19022.36	216.16	60.88		
	Col Sum	11357.38	14966.95	21759.47	33904.61	81988.40				
	Col Mean	86.04	113.39	164.84	256.85		**155.28**			
	Mobility Effect	-69.24	-41.90	9.56	101.57					

Table B.12: E-OP: Computation of Effects for Mega-bits Received

Shape	*Degree Level*	*Mobility (%)* 0	10	30	50	Row Sum	Row Mean	Row Effect	**Shape Effect**	**Degree Effect**
Square	Low	14.42	35.13	82.97	151.92	8533.20	71.11	-11.43	1.53	-17.72
	Medium	18.72	40.92	98.99	154.88	9405.57	78.38	-4.16		-0.03
	High	21.28	52.51	133.60	203.55	12328.42	102.74	20.19		17.74
C-Shape	Low	9.04	40.04	76.08	109.01	7025.07	58.54	-24.00	-1.53	
	Medium	25.32	56.70	106.26	158.34	10398.43	86.65	4.11		
	High	32.51	65.68	123.87	169.27	11739.84	97.83	15.29		
	Col Sum	3638.38	8729.61	18653.37	28409.18	59430.55				
	Col Mean	20.21	48.50	103.63	157.83		**82.54**			
	Mobility Effect	-62.33	-34.04	21.09	75.29					

Appendix C. Visual Tests for Validation of ANOVA & Linear Regression Assumptions

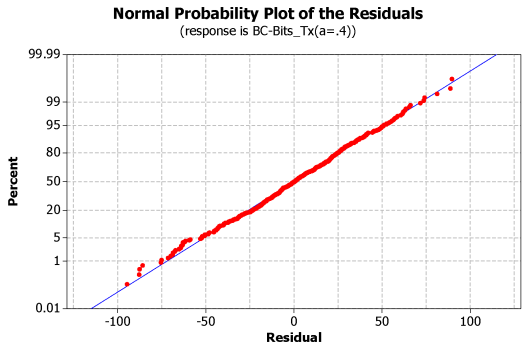

Figure C.1: MG-OP: Normal Plot of Residuals for BC-Bits Transmitted

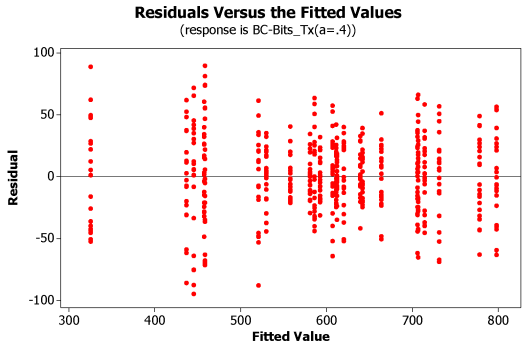

Figure C.2: MG-OP: Residuals vs. Fitted Values Plot for BC-Bits Transmitted

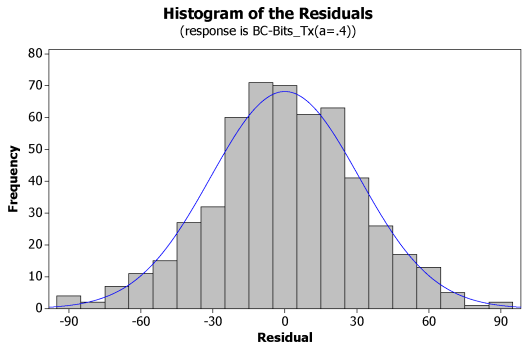

Figure C.3: MG-OP: Histogram of Residuals for BC-Bits Transmitted

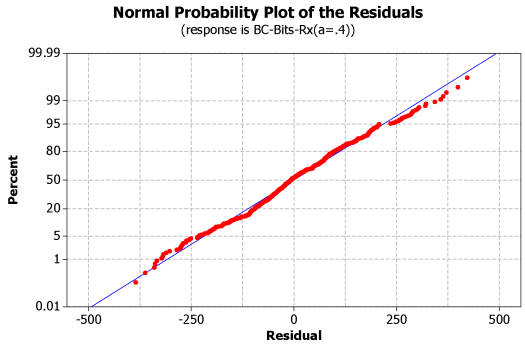

Figure C.4: MG-OP: Normal Plot of Residuals for BC-Bits Received

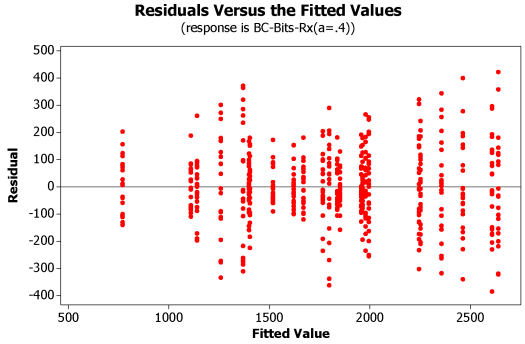

Figure C.5: MG-OP: Residuals vs. Fitted Values Plot for BC-Bits Received

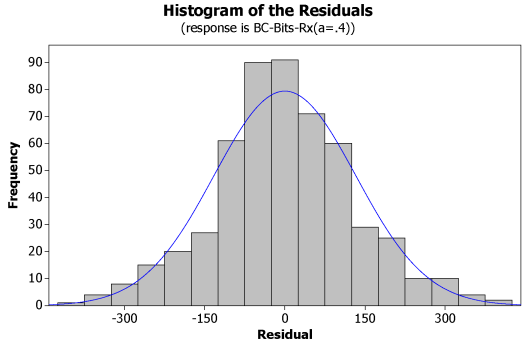

Figure C.6: MG-OP: Histogram of Residuals for BC-Bits Received

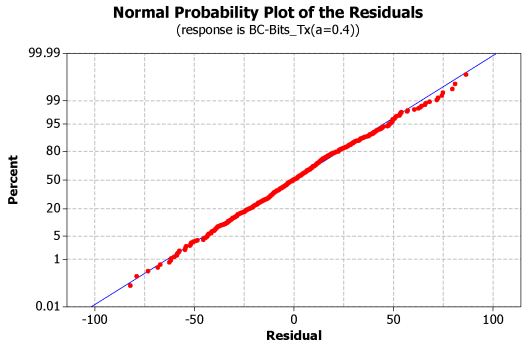

Figure C.7: E-OP: Normal Plot of Residuals for BC-Bits Transmitted

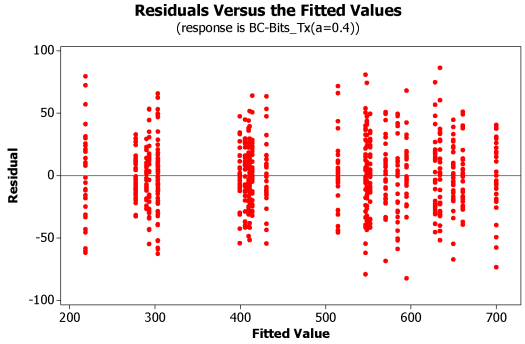

Figure C.8: E-OP: Residuals vs. Fitted Values Plot for BC-Bits Transmitted

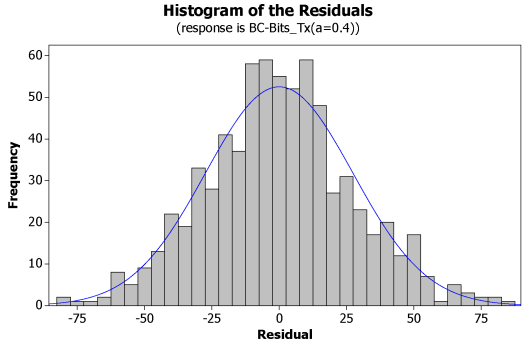

Figure C.9: E-OP: Histogram of Residuals for BC-Bits Transmitted

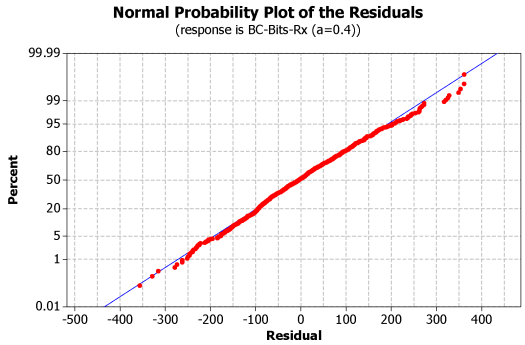

Figure C.10: E-OP: Normal Plot of Residuals for BC-Bits Received

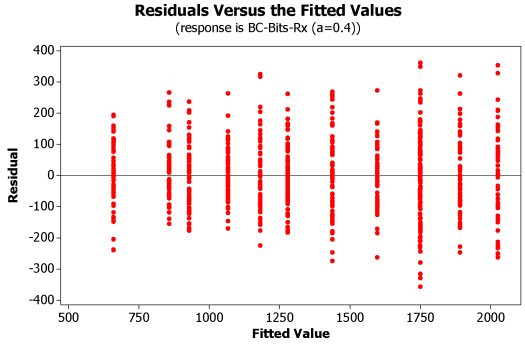

Figure C.11: E-OP: Residuals vs. Fitted Values Plot for BC-Bits Received

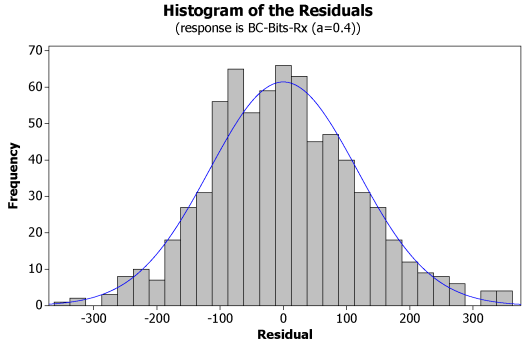

Figure C.12: E-OP: Histogram of Residuals for BC-Bits Received

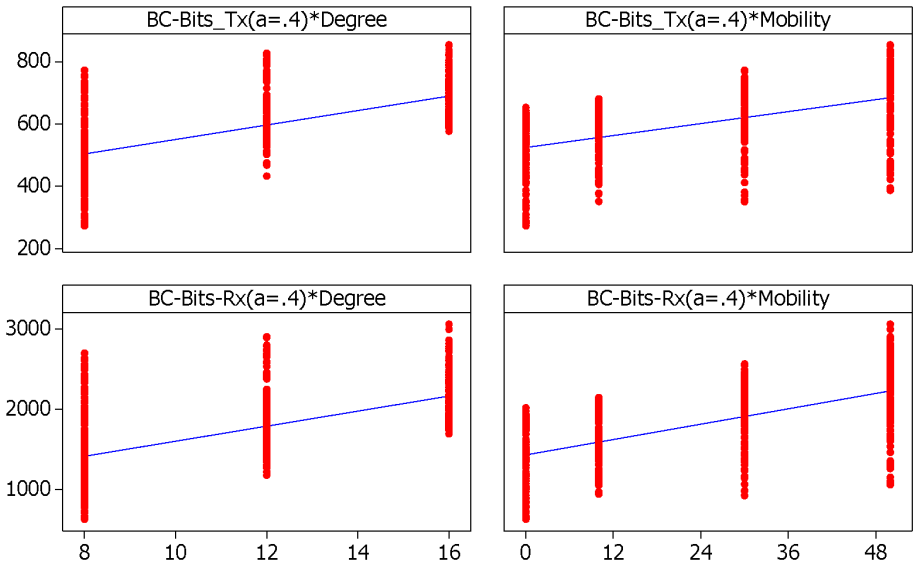

Figure C.13: MG-OP: Verification of Linear Relationship of BC-Bits Transmitted and Received vs. Degree and Mobility

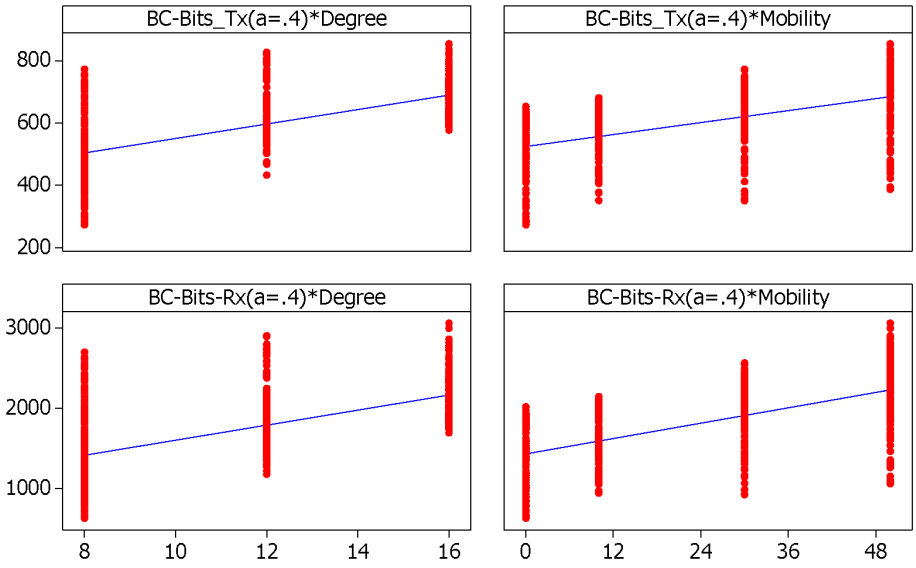

Figure C.14: MG-OP: Verification of Linear Relationship of BC-Bits Transmitted and Received vs. Degree and Mobility

Appendix D. Multiple Linear Regression Matrices

Table D.1: MG-OP: C-Matrix for BC-Bits Transmitted & BC-Bits Received

$$C = (X^T X)^{-1} = \begin{bmatrix} 0.0319562 & -0.0018939 & -0.0021307 & -0.0115562 \\ -0.0018939 & 0.0018939 & 0 & 0 \\ -0.0021307 & 0 & 0.0001776 & 0 \\ -0.0115562 & 0 & 0 & 0.0513611 \end{bmatrix} \tag{D.1}$$

Table D.2: E-OP: C-Matrix for BC-Bits Transmitted

$$C = (X^T X)^{-1} = \begin{bmatrix} 0.0359122 & -0.0083333 & -0.0015625 & -0.03518 & 0.05304 \\ -0.0083333 & 0.0055556 & 0 & 0 & 0 \\ -0.0015625 & 0 & 0.0001302 & 0 & 0 \\ -0.0351759 & 0 & 0 & 0.55939 & -1.03643 \\ 0.053043 & 0 & 0 & -1.03643 & 2.05891 \end{bmatrix} \tag{D.2}$$

Table D.3: E-OP: C-Matrix for BC-Bits Received

$$C = (X^T X)^{-1} = \begin{bmatrix} 0.0234122 & -0.0015625 & -0.03518 & 0.05304 \\ -0.0015625 & 0.0001302 & 0 & 0 \\ -0.0351759 & 0 & 0.55939 & -1.03643 \\ 0.053043 & 0 & -1.03643 & 2.05891 \end{bmatrix} \tag{D.3}$$

Bibliography

ASSC02. I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

BHE00. N. Bulusu, J. Heidemann, and D. Estrin. GPS-less lowcost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.

BHE01. N. Bulusu, J. Heidemann, and D. Estrin. Adaptive beacon placement. In *Proceedings of the 21st Annual International Conference on Distributed Computing Systems (ICDCS-21)*, April 2001.

BHET04. N. Bulusu, J. Heidemann, D. Estrin, and T. Tran. Self-configuring localization systems: Design and experimental evaluation. *Transactions on Embedded Computing Systems*, 3(1):24–60, 2004.

BM02. P. Bergamo and G. Mazzimi. Localization in sensor networks with fading and mobility. In *Proceedings of the 13th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC-13)*, September 2002.

Bra05. M. Brain. How motes work. http://computer.howstuffworks.com/mote5.htm, November 2005.

CHH01. S. Capkun, M. Hamdi, and J. Hubaux. GPS-free positioning in mobile ad-hoc networks. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*, 2001.

DFBT99. F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proceedings of IEEE International Conference on Robotics & Automations (ICRA'99)*, May 1999.

GKLP04. A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem. Distributed online localization in sensor networks using a moving target. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN-3)*, April 2004.

HB01a. J. Hightower and G. Borriello. Location sensing techniques. Technical report, University of Washington, Department of Computer Science and Engineering, Seattle, WA, July 2001.

HB01b. J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.

HE04. L. Hu and D. Evans. Localization for mobile sensor networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom-10)*, October 2004.

Jai91. R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling,*. Wiley-Interscience, New York, NY, May 1991.

JZ03. X. Ji and H. Zha. Robust sensor localization algorithm in wireless ad-hoc sensor networks. 2003.

JZ04. X. Ji and H. Zha. Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling. In *Proceedings of the 23rd Annual IEEE Conference on Computer Communications (INFOCOM-23)*, March 2004.

KMS+04. Y. Kwon, K. Mechitov, S. Sundresh, W. Kim, and G. Agha. Resilient localization for sensor networks in outdoor environments. In *Proceedings of 25th IEEE International Conference on Distributed Computing Systems (ICDCS-25)*, June 2004.

Lee05. P. Lee. Multivariate analysis. http://www-users.york.ac.uk/∽pml1, April 2005.

LH05. H. Lim and J. Hou. Localization for anisotropic sensor networks. In *Proceedings of the 24th Annual IEEE Conference on Computer Communications (INFOCOM-24)*, March 2005.

LR03. K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, 43(4):499–518, 2003.

LSS04. X. Li, H. Shi, and Y. Shang. A map-growing localization algorithm for ad-hoc wireless sensor networks. In *Proceedings of 10th International Conference on Parallel and Distributed Systems (ICPADS-10)*, July 2004.

LWH04. C. Liu, K. Wu, and T. He. Sensor localization with ring overlapping based on comparison of received signal strength indicator. In *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS-1)*, October 2004.

MSK04. M. Minami, S. Saruwatari, and T. Kashima. Implementation-based approach for designing practical sensor network systems. In *Procedings of 11th Asia-Pacific Software Engineering Conference(APSEC-11)*, December 2004.

NN01. D. Niculescu and B. Nath. Ad-hoc positioning system. In *Procedings of the 44th Annual IEEE Global Communications Conference (GLOBCOM-44)*, November 2001.

NN03a. D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AOA. In *Proceedings of the 22nd Annual IEEE Conference on Computer Communications (INFOCOM-22)*, March 2003.

NN03b. D. Niculescu and B. Nath. DV based positioning in ad hoc networks. *Telecommunication Systems*, 22(1-4):267–280, 2003.

PBDT05. N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. In *Proceedings of the 24th Annual IEEE Conference on Computer Communications (INFOCOM-24)*, March 2005.

PCB00. N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket compass for context-aware mobile applications. In *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom-6)*, July 2000.

PH04. N. Patwari and A. Hero. Manifold learning algorithms for localization in wireless sensor networks. In *Proceedings of the 6th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-6)*, May 2004.

Ric05. A. Ricadela. Sensors everywhere. Information Week, ArticleID=57702816, http://www.informationweek.com/, January 2005.

SBM+04. J. Sallai, G. Balogh, M. Maróti, Á. Lédeczi, and B. Kusy. Acoustic ranging in resource-constrained sensor networks. In *Proceedings of the International Conference on Wireless Networks (ICWN)*, June 2004.

SHS01. A. Savvides, C. Han, and M. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom-7)*, July 2001.

SHS04. R. Stoleru, T. He, and J. Stankovic. Walking GPS: A practical solution for localization in manually deployed wireless sensor networks. In *Proceedings of the 1st IEEE Workshop on Embedded Networked Sensors (EmNetS-1)*, November 2004.

SPS02. A. Savvides, H. Park, and M. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA-1)*, September 2002.

SR04. M. Sichitiu and V. Ramadurai. Localization of wireless sensor networks with a mobile beacon. In *Proceedings of the 1st IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS-1)*, October 2004.

SRB01. C. Savarese, J.M Rabaey, and J. Beutel. Locationing in distributed ad hoc wireless sensor networks. In *Procedings of the 3rd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-3)*, May 2001.

WG97. P. Wolf and C. Ghilani. *Adjustment Computations: Statistics and Lease Squares in Surveying and GIS*. Wiley-Interscience, New York, NY, 1997.

Whi02. K. Whitehouse. *The Design of Calamari: an Ad-hoc Localization System for Sensor Networks.* Ph.D. thesis, University of California at Berkeley, Los Angeles, California, Fall 2002.

Xbo05. Crossbow Technology Incorporated. http://www.xbow.com/, July 2005.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
23–03–2006	Master's Thesis	Sept 2004 — Mar 2006

4. TITLE AND SUBTITLE

Evaluation and Analysis of
Node Localization Power Cost in
Ad-hoc Wireless Sensor Networks with Mobility

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

BRIAN A. SESSLER, 1Lt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

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

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Bldg 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GCE/ENG/06-07

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

AFRL/IFSC (AFMC)
Mr. William Koenig
2241 Avionics Circle
WPAFB OH 45433-7765
DSN: 785-4709x3172
William.Koenig@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

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

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT One of the key concerns with location aware AWSNs is how sensor nodes determine their position. The inherent power limitations of an AWSN along with the requirement for long network lifetimes, makes achieving fast and power-efficient localization vital. This research examines the cost (in terms of power) of network irregularities on communications and localization in an AWSN. The number of data bits transmitted and received are significantly effected by varying levels of mobility, node degree, and network shape. The concurrent localization approach, used by the APS-Euclidean algorithm, has significantly more accurate position estimates with a higher percentage of nodes localized, while requiring 50% less data communications overhead than the Map-Growing algorithm. Analytical power models capable of estimating the power required to localize are derived. The average amount of data communications required by either these algorithms, in a highly mobile network with a relatively high degree, consumes less than 2.0% of the power capacity of an average 560mA-hr battery. This is less than expected and contrary to the common perception that localization algorithms consume a significant amount of a nodes power.

15. SUBJECT TERMS

ad-hoc networks, sensor networks, localization, positioning

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Rusty O. Baldwin (ENG)
U	U	U	UU	119	19b. TELEPHONE NUMBER *(include area code)* (937) 255–6565 x4445, rusty.baldwin@afit.edu