Distributed Localization for Wireless Distributed Networks in Indoor Environments

Hermie Pante Mendoza

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical and Computer Engineering

Jeffrey H. Reed, Chair
Tamal Bose
Haris I. Volos

June 28, 2011
Blacksburg, Virginia

Keywords: wireless distributed computing networks, localization, indoor localization, fingerprinting

Distributed Localization for Wireless Distributed Networks in Indoor Environments

Hermie Pante Mendoza

(ABSTRACT)

Positioning systems enable location-awareness for mobile devices, computers, and even tactical radios. From the collected location information, location-based services can be realized. One type of positioning system is based on location fingerprints. Unlike the conventional positioning techniques of time of or time delay of arrival (TOA/TDOA) or even angle of arrival (AOA), fingerprinting associates unique characteristics such as received signal strength (RSS) that differentiates a location from another location. The location-dependent characteristics then can be used to infer a user's location. Furthermore, fingerprinting requires no specialized hardware because of its reliance on an existing communications infrastructure.

In estimating a user's position, fingerprint-based positioning systems are centrally calculated on a mobile computer using either a Euclidean distance algorithm, Bayesian statistics, or neural networks. With large service areas and, subsequently, large radio maps, one mobile computer may not have the adequate resources to locally compute a user's position. Wireless distributed computing provides a *means* for the mobile computer to meet the location-based service requirements and increase its network lifetime. This thesis develops distributed localization algorithms to be used in an indoor fingerprint-based positioning system. Fingerprint calculations are not computed on a single device, but rather on a wireless distributed computing network on Virginia Tech's Cognitive Radio Network Testbed (CORNET).

Dedication

"The cure for all things is salt water: sweat, tears, or the sea." -Isaak Dinesen

For my family, doggies, and the late Dr. Keith "Doc" Gross, my undergraduate advisor at the United States Coast Guard Academy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background Information

Before the ubiquity of Global Positioning System (GPS) satellites and wireless communication technology, man primarily relied solely on the celestial bodies to navigate both land and sea for centuries [1]. By the Age of Exploration, explorers could navigate the oceans with increased confidence in the advancements of celestial navigation, cartography, and astronomy. The greatest challenge for explorers and navigators was in determining longitude, the geographic coordinate that specifies the east-west position of a point on the Earth's surface. Latitude, the angular distance of a location in reference to the equator, was determined relatively easily by measuring the sun's altitude at noon with the aid of a table listing the sun's daily declination [2]. Thus, the key to successfully determining longitude was the recognition of its direct relationship to accurate time-keeping. With the time of a fixed reference point and the local time of an event, the difference between the reference and apparent local time would give a position relative to the fixed location. In 1773, the British Board of Longitude awarded John Harrison for inventing the marine chronometer, which was based on the lunar distance method. By late 19th century, Harrison's invention had widespread usage [3]. By 1904, marine radio signals were broadcast to vessels to correct chronometer errors, which eventually evolved to broadcasting marine radiobeacons. The 20th century innovations of Decca, long range navigation system (LORAN), GPS, and wireless cellular technologies not only enabled accurate global and local position awareness, but their success often relied on accurate clock synchronization.

One could easily see that the history of navigation closely coincided with development of world economics and military conquest. Specifically, the accurate time-keeping and advances in radio communications correlated geographic information with an event or object. Though the significance of this information would be first reaped for military objectives, position location based services have become ubiquitous and are widely used by public and private

sectors for various applications. For example, in the United States, the Federal Communications Commission (FCC) has mandated in the Enhanced 911 (E-911) service that all cellular operators to be able to estimate the position of an emergency caller with an accuracy of less than 125 meters [4]. Position information enables multiple other applications including road-side assistance, location-sensitive billing, and farming.

1.2 Position Location for Wireless Distributed Networks

As related technologies, both wireless sensor networks and ad-hoc networks continue to enable new applications in monitoring and control. Examples include wildlife habitat monitoring, intrusion detection, and retina prosthesis. The enabling technology for promising applications is the continuing improvement of digital circuitry integration, which miniaturizes the node's form-factor and reduces production cost.

As a result of reducing form-factor, the node's computing capability and lifetime can become limited. Consequently, networks must jointly optimize essential data collaboration amongst nodes and minimize communication traffic, at a minimum. Networks must also satisfy, if necessary, stringent quality-of-service (QoS) and security requirements. To meet all of these objectives, clusters of nodes can form distributed networks and harness the benefits of collaborative computing in a distributed manner. The concept of distributed computing (DC) has been well-researched for wired networks, but DC has not been applied to the wireless networks. Note that the term wireless distributed computing (WDC) will be used to distinguish between the two types of DC. Because of the complexities of the wireless channel, DC cannot be directly implemented [5].

The meaningfulness of the gathered network data is realized when it is coupled with the geographical position of the deployed nodes. Many wireless network applications will assume a reasonably precise position for static nodes, which are often pre-positioned using a location service such as GPS. However, GPS is not always a suitable location service for low-cost, resource-constrained nodes in a WDC network. Location solutions implemented in robotics and ubiquitous computing demand significant computing power and are not always energy efficient.

In this thesis, we propose a distributed localization solution specifically for WDC networks. The sometimes extreme environments, where WDC nodes are deployed, lack pre-existing communications infrastructure and, in many cases, may not have access to GPS motivate the reasoning why a flexible distributed position system and scheme are needed. Another motivator for using a distributed localization algorithm is to reduce the necessity of centralized computations of network wide positioning problems. The lack of a centralized computations of global positions inherently promotes security while extending the network lifetime by efficiently using the computational capabilities of available nodes.

We note that the term *distributed* for this research will denote events, applications, services, etc. that are executed concurrently, with separate parts running simultaneously on independent platforms. This definition is similar in spirit to what is currently applied in the field of computer science.

1.3 Problem Statement: The Focus of this Work

WDC networks have a natural need for position information, but not all positioning algorithms are applicable. The primary goal of this work is to develop a prototype locationing subsystem capable of providing position estimates of nodes within a WDC network. The algorithm driving the locationing service should be robust to any WDC application, but there may be special instances where the distributed location solution might not be applicable. Consequently, implementation recommendations will be made at the algorithmic and network levels. Metrics used to judge the effectiveness of the distributed location algorithm will be the expected error margins in various scenarios, flexibility, and robustness for special cases.

1.4 Terminology Used Throughout This Report

In a WDC network, there are many variables relevant for positioning. This section will briefly describe the conventions that will be used in the remainder of the thesis.

The network will consist of two types of nodes: anchor nodes and blind nodes (often referred to as simply 'nodes'). Note that the functionality of a blind node is not required information for any of the position location algorithms. Anchor nodes, in comparison to blind nodes, have *a priori* knowledge of their position within the global frame of reference. To accomplish this, for example, anchor nodes could be equipped with GPS receivers or some means to obtain their actual position. Anchor nodes effectively create reference positions within a network and provide a system of reference for entities external to it.

Regardless of the network configuration, a network is characterized by the following information:

Table 1.1: Network Characteristics

n	The total number of nodes in the network
a	The number of anchor nodes of the n nodes
D	The space dimension in which the network resides (i.e. 2-D or 3-D)

The analysis of the distributed algorithms will often focus on error magnitude of position estimates. This numeric quantity is given as a percentage of error and determined by the magnitude differences between a node's true and estimated position. The true position is the absolute location of a node within a global reference, while the estimated position is the perceived or "guessed" node location.

1.5 Overview of Thesis

The primary focus of this thesis is the advantages and disadvantages of several algorithms used to localize nodes in a WDC network. A major assumption of this work that a subsystem will provide range measurements or information as inputs to the algorithms, which will be noisy. With the range measurements, the algorithms derive estimated positions for each node in the WDC network, and then pass their coordinates to network.

Chapter 2 will discuss the localization techniques currently used in wireless sensor networks (WSN), ad-hoc networks, and wireless networks. Topics covered include the theory of multilateration, methods of obtaining range measurements, and other relevant work. Chapter 3 will develop the background information of WDC networks, especially highlighting its differences from traditional DC networks. Chapter 4 will focus on the developed distributed position algorithms specifically for an indoor WDC network and the experimentation used to verify them. Chapter 5 will highlight implementation details of the algorithms developed in Chapter 4 including in depth assessment of algorithmic costs and the merits and disadvantages of them. Chapter 6 will describe how the distributed algorithms perform in the WDC-based indoor positioning system. Lastly, Chapter 7 will conclude this report and and propose areas for improvement and future growth.

Chapter 2

Positioning Approaches and Techniques

This chapter provides background information in the study of position location, or commonly referred to incorrectly as geolocation. What is first discussed in Section 2.1 is the mathematical interpretation of positioning, specifically lateration and its derivatives. Since the mathematical calculations are dependent on a node obtaining range measurements, the different methods of obtaining range measurements are highlighted in Section 2.2. Lastly, the concept of range-free position measurements and techniques are discussed.

2.1 The Principle of Positioning

The concept of position location and geolocation are closely related, but are distinguished by their emphasis on a set or network coordinates or a real-world location. The remainder of the report focuses on position location. Any desire to correlate network coordinates to a physical location may require additional hardware for nodes.

The foundations of position location are very dependent on a predefined frame of reference. Thus, all positions within are only relative. For example, suppose a postman is to deliver a package to a customer in a house at a specific location. The house's location does not provide insightful information as what city, state, or even country where the customer resides. Without a meaningful reference coordinate system, the house's location is useless, and the package could not be delivered efficiently. Therefore, it is assumed that there exists some global coordinate system available as reference to which the anchor nodes are associated. Details of the global coordinate system are application-specific and are irrelevant for a position algorithm. Associating global coordinates with an algorithm's coordinate system is, in fact, a part of geolocation procedures. These methods will not be highlighted. Consequently, the primary goal of any position location algorithm is to determine the node's position within a

global coordinate system. This can be accomplished with trilateration or multilateration.

2.1.1 Trilateration

Trilateration, a more specialized version of triangulation, is one method of acquiring a point's location and forms the geometric foundation for the algorithms presented here. The core idea of lateration is that the coordinates of a point of interest (POI) can be determined by calculating the lengths, or edges, between objects. The POI has a unique coordinate when there exists, at minimum, three reference points associated with it, forming a triangle in the two-dimensional (2-D) space. Note that in the three-dimensional space, a POI will need at least four reference points to form a tetrahedron. Once the edges and angles of the polygon are measured, the POI's can be localized.

To better illustrate this concept, please refer to Figure 2.1. Consider $\mathbb{R}^2$ with POI m with unknown coordinates. Surrounding m are three reference points, a_1, a_2, and a_3 with known coordinates. Suppose the POI m estimates the distance r_1 from reference a_1, m can conclude that it is located on the perimeter of a circle, centered at a_1 and radius r_1. The POI m will then measure distances r_2 to reference a_2 and, again, concludes that it is located on the perimeter of a circle, centered at a_2 and radius r_2. Note that the presence of another circle implies that this new circle intersects the first circle, centered at a_1, at two points. Lastly, m will determine distance r_3 to references a_3. With these three distances, m can conclude that it is located at the intersection of three separate circles. A final comment to note is that when extending lateration to $\mathbb{R}^3$, the circles are replaced as spheres. The intersection of spheres, a convex 3-D polygon, become the coordinates of m.

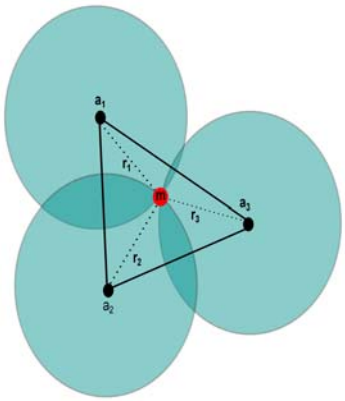

Figure 2.1: Geometric Interpretation of Triangulation

Mathematically, the estimated distances (r_i) and the known positions of the references points

(x_i, y_i) form the following system of non-linear equations:

$$\begin{aligned} (x_1 - x)^2 + (y_1 - y)^2 &= r_1^2 \\ (x_2 - x)^2 + (y_2 - y)^2 &= r_2^2 \\ (x_3 - x)^2 + (y_3 - y)^2 &= r_3^2 \end{aligned} \tag{2.1}$$

where the unknown position of m is denoted by (x, y). These equations can be linearized by subtracting the last equation from the first two equations.

$$\begin{aligned} x_1^2 - x_3^2 - 2(x_1 - x_3)x + y_1^2 - y_3^2 - 2(y_1 - y_3)y &= r_1^2 - r_3^2 \\ x_2^2 - x_3^2 - 2(x_2 - x_3)x + y_2^2 - y_3^2 - 2(y_2 - y_3)y &= r_2^2 - r_3^2 \end{aligned} \tag{2.2}$$

From Equation (2.2), reordering the terms will yield a system of linear equations in matrix form $Ax = b$, where

$$\begin{aligned} A &= \begin{bmatrix} 2(x_1 - x_3) & 2(y_1 - y_3) \\ 2(x_2 - x_3) & 2(y_2 - y_3) \end{bmatrix}, \\ b &= \begin{bmatrix} x_1^2 - x_3^2 + y_1^2 - y_3^2 - d_3^2 - d_1^2 \\ x_2^2 - x_3^2 + y_2^2 - y_3^2 - d_3^2 - d_2^2 \end{bmatrix} \end{aligned} \tag{2.3}$$

In the case of n reference points, the system of non-linear equations become:

$$\begin{aligned} (x_1 - x)^2 + (y_1 - y)^2 &= r_1^2 \\ (x_2 - x)^2 + (y_2 - y)^2 &= r_2^2 \\ &\vdots \\ (x_n - x)^2 + (y_n - y)^2 &= r_n^2. \end{aligned} \tag{2.4}$$

Then the linearized equations in matrix form is:

$$\begin{aligned} A &= \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ 2(x_2 - x_n) & 2(y_2 - y_n) \\ \vdots & \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix}, \\ b &= \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 - d_n^2 - d_1^2 \\ x_2^2 - x_n^2 + y_2^2 - y_n^2 - d_n^2 - d_2^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 - d_n^2 - d_{n-1}^2 \end{bmatrix}. \end{aligned} \tag{2.5}$$

To solve Equation (2.5), the least squares (LS) algorithm is used to derive an estimate of m's position: $\hat{x} = \left(A^T A\right)^{-1} A^T b$. More information about the LS algorithm will discussed in further detail in Section 4.4.

2.1.2 Multilateration

In comparison to trilateration or triangulation, the geometry of multilateration is more complex. Rather than using triangles, hyperbolas are formed from computing the time difference of arrival (TDOA) of a signal emitted from an object to at least three reference points. This concept is best described with the help of Figure 2.2.

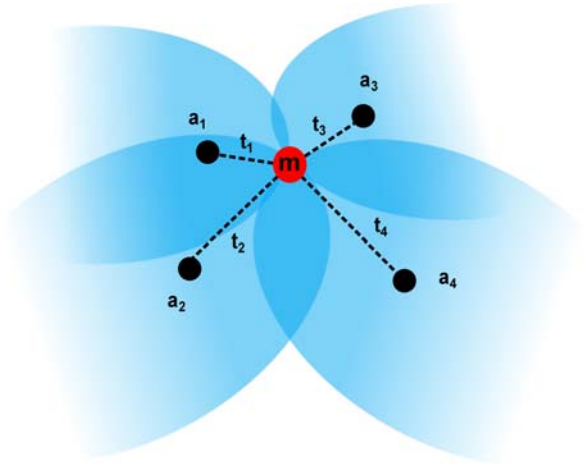

Figure 2.2: Geometric Interpretation of Multilateration

Suppose that reference points a_1 through a_4 have known positions with non-equal distances from m. The POI m will emit a signal to all the reference points a_i, but the signal will arrive at different times, denoted t_i. With only two reference points and one TDOA, m's estimated location is on a hyperboloid. With three reference points, the intersection is a curve on two hyperboloids. This would be sufficient for m's position if in $\mathbb{R}^2$. Note that reference point a_4 is only needed to fully determine m's position in $\mathbb{R}^3$.

The time of arrival (TOA) of the signal from m to each reference point is calculated by dividing their distance from m by the signal propagation rate, denoted by (c). For the scenario in Figure 2.2, the TOA equations are:

$$\begin{aligned} t_1 &= \tfrac{1}{c}\sqrt{(x_1-x)^2+(y_1-y)^2+(z_1-z)^2} \\ t_2 &= \tfrac{1}{c}\sqrt{(x_2-x)^2+(y_2-y)^2+(z_2-z)^2} \\ t_3 &= \tfrac{1}{c}\sqrt{(x_3-x)^2+(y_3-y)^2+(z_3-z)^2} \\ t_4 &= \tfrac{1}{c}\sqrt{(x_4-x)^2+(y_4-y)^2+(z_4-z)^2} \end{aligned} \tag{2.6}$$

If considering $\mathbb{R}^2$, the TOA equations are notably reduced by eliminating the $(z_i - z)$ terms.

Then the TDOA equations are shown to be, with reference point a_1 as the origin point,

$$\begin{aligned}
\tau_2 = t_2 - t_1 &= \tfrac{1}{c}\left(\sqrt{(x_2'-x)^2+(y_2'-y)^2+(z_2'-z)^2} - \sqrt{(x)^2+(y)^2+(z)^2}\right) \\
\tau_3 = t_3 - t_1 &= \tfrac{1}{c}\left(\sqrt{(x_3'-x)^2+(y_3'-y)^2+(z_3'-z)^2} - \sqrt{(x)^2+(y)^2+(z)^2}\right) \\
\tau_4 = t_4 - t_1 &= \tfrac{1}{c}\left(\sqrt{(x_4'-x)^2+(y_4'-y)^2+(z_4'-z)^2} - \sqrt{(x)^2+(y)^2+(z)^2}\right)
\end{aligned} \tag{2.7}$$

where (x_i', y_i', z_i') is the location of reference point i with respect to the origin a_1. Each individual equation in Equation (2.7) is a separate hyperboloid.

Because of the fixed physical relationship between the speed of the signal and distance, multilateration often results in greater accuracy in localizing a POI. This significant advantage to trilateration requires a unified clock reference amongst all reference points, which may require additional hardware, to calculate TDOA's.

This section explained the mathematics behind the primary position location techniques of trilateration and multilateration. Note that trilateration is a specialization of triangulation and only relies on distance measurements between the POI and reference points. In comparison, multilateration requires a TDOA calculations between reference points to form hyperboloids, which results in a greater accuracy in the POI's coordinates.

2.2 Range-Based Measurements

Trilateration's dependence on the distances between the POI and the reference points is one of its weaknesses. The distance between the POI and a reference point is often called a *range*. This section will focus on the techniques used to obtain raw range measurements: angle of arrival (AOA), TDOA/TOA, and received signal strength indicator (RSSI).

Range-based measurements can be AOA, RSS, TDOA/TOA, or a hybrid combination. In this subsection, these three major techniques are described. Then the strengths and weaknesses of each are summarized.

2.2.1 AOA

Although the AOA technique is not used in in this report, its significance in position location should not be neglected. The premise of AOA is to determine the direction, or bearing, of a radio signal incident on an antenna array. To accomplish this, the TDOA of a signal is measured at each individual element on the antenna. This can be considered the reverse of antenna beamforming. When combined with distance and/or additional angle measurements, a POI's position can be derived. Other techniques to calculate AOA are discussed in [6]

and [7]. A major disadvantage of AOA is the requirement of specialized antenna arrays. Depending on the application, antenna arrays could be impractical or even costly.

2.2.2 RSSI

RSSI uses the relative signal strength between receivers at known locations and a transmitter for localization. If known, the propagation loss and transmitter power can improve the accuracy of the estimated distances.

In general, attenuation of signal strength is caused by one of three independent factors: (i) path loss, (ii) multipath fading, and (iii) shadowing [8]. Here, a brief review of the attenuation factors is given below.

- The path loss factor n characterizes the rate at which the signal power decays as distance d increases between the receiver and the transmitter. In free space, $n = 2$ is used because the signal power is proportional to d^{-2}. Environments subjected to reflection and deflection (i.e. floors, walls, and buildings) observe a path loss factor of $n > 2$. For example, in urban areas, $n = 4$ is often used in channel models [9].

- Multipath fading, sometimes referred to fast fading, is the rapid change of the complex envelope of the received signal in multipath propagation, caused by the reception of multiple copies of a transmitted signal. If there exists a dominant component among the multiple copies, the amplitude distribution can be described by a Rician distribution. The Rayleigh distribution occurs when the real (R) and imaginary (I) components of the complex envelope have zero means and a common variance with amplitude $\sqrt{R^2 + I^2}$. If R or I or both have non-zero mean yet a common variance, the amplitude distribution is Rician.

- Shadowing, or slow fading, is the slow variation in the received signal strength due to scatters, or obstacles, in the propagation paths. In nature, log-normal shadowing is widely present [10].

Given the output power of the transmitting antenna P_t, the gain of the receiving and transmitting antennas G_r and G_t, and the received power P_r, the location of the transmitter can be determined. The equation shown below (Equation 2.8) can be solved for the distance d, which is an estimate between the transmitter and receiver.

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi d} \right)^2 \tag{2.8}$$

Note that Equation (2.8) assumes using the free space model and a path loss exponent of 2. To express signal power in dBm, Equation (2.8) can be converted into

$$P_r\,[dBm] = P_t + 10\log G_t + 10\log G_r + 20\log\lambda - 20\log 4\pi - 20\log d \tag{2.9}$$

where P_t is in dBm.

2.2.3 TOA and TDOA

Unlike the previous techniques discussed, TOA, or time of flight (TOF), uses the measured time for a signal to propagate between a receiver and transmitter. With the signal's propagation speed, distance can be directly estimated. Then TDOA, round-trip time of flight (RTOF), is based on the time difference of a signal's arrival. Systems that use TDOA include LORAN and GPS [11].

To use either TOA or TDOA, network synchronization is required. Inaccuracies in oscillator synchronization in either the receiver or transmitter directly translates to inaccuracies in an estimated position.

Table 2.1 summarizes the merits and weaknesses of the discussed range-based measurements. Note that the level of accuracy of the POI increases as one moves left to right in the table. Most notably, RSSI measurements frequently yield poor distance estimates [12]. Better distances estimates are obtained using TOA, especially combining acoustic and RF signals. Concurrently, the computational and hardware complexity increases as well.

For this report, RSSI measurements are assumed to be the method used to generate range measurements. However, the position location algorithms discussed can be used for any of the aforementioned techniques. What is of concern for a position location algorithm is the expected error distributions associated with range estimates derived from these techniques.

2.3 Range-Free Position Estimation

This next section explains an alternative to range-based position location called *range-free* schemes. Unlike the former, range-free schemes makes no assumption of the validity or even availability of range or angle measurements. What is assumed by range-free schemes is their deployment in networks with $N >> 4$ and that the POI is indeed a node of interest (NOI). First discussed are the most common methods of obtaining a range-free measurements. Then we look at their associated position estimation techniques.

Table 2.1: Comparison of Range Based Measurements

	AOA	**RSSI**	**TOA**	**TDOA**
Observed Information	Incident angle of received signal	Received signal strength	Propagation time of received signal	Difference of arrival times of signal emitted from a source
Suitable Environment	Rural	Urban, Indoor	Urban, Suburban	Urban, Suburban
Example Application	Phase array radar	RFID tags	Satellite positioning systems	Navigation-based positioning
Advantages	• Passive system • Synchronization not required	• Low cost • Synchronization not required • Easy implementation	• High accuracy with coherent detection	• High accuracy with coherent detection
Disadvantages	• Requires specialized hardware • Low accuracy • Susceptible to radio environment changes	• Difficult to achieve high accuracy of TOA/TODA • Susceptible to radio environment changes	• Requires duplex transmission • Requires network synchronization	• Costly computational calculations • Requires network synchronization

2.3.1 Sum-Dist

Savides, et. al in [14] describe a simple approach in determining distances to anchor nodes-simply add the ranges at each hop during a network flood. This process begins with the anchor nodes, broadcasting a message that includes their identity, position, and the path length set to 0 to nodes within a certain distance. Once a node receives this message, it will add the measured range to the path length only if the current path length is less than the previous one. If this is indeed true, the node broadcasts the message to other nodes. Another constraint in the node forwarding message is if the flood limit is not exceeded. A flood limit describes the maximum number of times a message can be forwarded. Thus, each node in the network will have the position and minimum path length to anchor nodes within flood limits.

The sum-dist process is susceptible to range errors in multiple-hop scenarios. Furthermore, the accumulated range errors become exacerbated in large networks with a limited number of anchor nodes or have poor ranging hardware.

2.3.2 DV-Hop and Hop-Terrain

The drawbacks of sum-distance inspired the development of DV-Hop [6] and Hop-Terrain [15]. In both DV-Hop and Hop-Terrain, each use the topological information of hop count instead of summing ranges. For simplicity, we will now refer to both as DV-Hop. This process consists of two phases. The first phase is similar to sum-dist, where each node has the position and minimum hop count to the reference nodes within a flood limit. In the second phase, a second broadcast message is sent to convert the hop counts into distances. Distance conversions are computed by multiplying the hop count with an average hop distance. For example, reference node a_1 estimates the position of neighboring reference node a_2 in the first phase of DV-Hop, computes the distance between them, and then divides this distance by the number of hops separating them. This final result is the average hop distance.

2.3.3 Range-Free Techniques

Although trilateration and multilateration are used in conjunction with range-based measurements, range-free measurements occasionally use these techniques. Since the general idea in applying trilateration or multilateration is analogous in the range-free scenario, this subsection will highlight the other techniques used.

Min-Max

An alternative to using trilateration or multilateration is to apply the min-max algorithm (sometimes referred to as the minimax algorithm) [13]. Its main premise is to to create a square bounding box for each reference node from its position and distance estimate. The NOI or POI is calculated as the center of the intersection formed by the anchor bounding boxes. This is very much in the spirit of lateration, except the circles are replaced with squares.

To create a bounding box around reference point i, its estimated distance (d_i) is subtracted from its anchor position (x_i, y_i). Then bounding box's dimensions are:

$$[x_i - d_i, y_i - d_i] \times [x_i + d_i, y_i + d_i] \tag{2.10}$$

Once all bounding boxes are determined, the intersection's dimensions are computed by

$$\max_i (x_i - d_i, y_i - d_i) \times \min_i (x_i + d_i, y_i + d_i) \tag{2.11}$$

Then the POI is the average of both corner coordinates, or the center of the bounding boxes.

Centroid

This technique uses N anchor nodes with known coordinates $(x_i, y_i), i = 1, \ldots, N$ [16]. These anchor nodes will transmit to the NOI their coordinates, which is also referred to as *seeds*. Then the NOI estimates its position as the centroid of these points. Specifically, the centroid's coordinates is calculated as:

$$(\hat{x}, \hat{y}) = \left(\frac{1}{N} \sum_{i=1}^{N} x_i, \sum_{i=1}^{n} y_i \right) \tag{2.12}$$

If the anchor nodes are poorly positioned or do not exist in the vicinity of the NOI, localization error may increase [17]. Unfortunately, this is often the case in ad-hoc network deployments.

Approximate Point-In-Test (APIT)

Unlike the Centroid technique, the anchor nodes in APIT transmit beacons to the NOI. The geographical environment is isolated into triangular regions between all the anchor nodes [18]. If a NOI resides inside or outside a specific triangular region, the NOI can constrain the area it believes it is in. The NOI, in addition, can further reduce this area when using all possible combinations of anchor node positions.

To narrow the possible area a NOI resides, it employs the point-in-triangulation (PIT) test. The test consists of the NOI choosing three "audible" anchor nodes and determining whether

it is inside or outside the triangle formed by the anchor nodes. APIT builds on the foundation of PIT, but it uses all combinations of audible anchor nodes to achieve higher accuracy. The estimated position of the NOI is determined by calculating the center of gravity (COG) of the intersection of all the feasible triangle areas. Thus, APIT consists of four steps: (1) beacon exchange among anchor nodes, (2) PIT testing, (3) triangle aggregation, and (4) COG calculation.

Undoubtedly, the most important step in APIT is step (2), determining if the NOI is indeed inside a triangle formed by three audible anchor nodes. Step (2) of APIT becomes increasingly critical if the NOI is mobile. Consider the case when an NOI is inside the triangle formed by anchor nodes A, B, and C (refer to Figure 2.3). Theoretically, if the NOI shifts in any direction inside the triangle, the new position will be nearer or further away from at least one of the anchor nodes A, B, or C. If the NOI moves outside the triangle, there will exist a direction of departure that is closest or furtherest from all the anchor nodes simultaneously. Since the PIT test is computationally intensive, an approximate PIT test is implemented in practice.

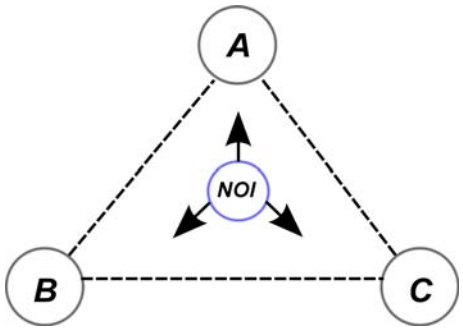

Figure 2.3: PIT Test

In comparison to the standard PIT test, the approximate PIT test uses the assumption that in a certain propagation direction, within a narrow angle from a transmitting anchor node, the RSS decreases monotonically with distance. Movement of the NOI is simulated by exchanging anchor neighbor information via beaconing. Thus, the NOI can more easily assume either it is inside or outside triangle ABC. The approximate PIT test is susceptible to errors if the NOI is located on the edge of the triangle ABC or an improbable location.

Step (3) of APIT aggregates the results of each individual PIT test onto a grid. Specifically, the network area is divided smaller square regions to form a grid array, which represents the maximum area in which a NOI will likely reside. If a PIT test determines that the NOI is inside a triangle, the grid regions corresponding to the triangle is incremented. Otherwise, the NOI is outside the triangle, and this grid region is decremented. Computing the COG, step (4), occurs once all PIT test results are aggregated and mapped onto the grid array. The maximum overlapping area (i.e. the squares with the largest counters) is used to calculate the COG.

APIT is very simple to implement in practice. Yet its major drawback is that RSS is susceptible to large- and small-fading, which will increase PIT test failure rates. Furthermore, the anchor node density of the network, as in the centroid scheme, may result in not enough overlapping triangular regions and decrease position accuracy.

Scene Analysis

In scene analysis, it focuses on examining a view from a particular point of view to estimate an observer's location [19], [20]. This technique detects movement between frames or time instances with differential analysis. Here, a scene can either be an actual visual image or mapped electromagnetic characteristics.

Consider a scenario where a NOI is seen by several anchor nodes. Each location n_o that the NOI passes through will have an associated vector or location signature, $F_o = (F_{o1}, \ldots, F_{0n})$ where F_{on} may denote the field intensity associated with an anchor node i. Once all NOI observations are obtained, they will be compared against the reference signatures and the closest reference will be used to estimate the NOI's location.

The ideal mapping $n_o \leftrightarrow F_o$ may heavily depart from practical observations due to clutter, propagation impairments, and low instrument resolution. Thus, the signature must be adapted on the basis of multiple observations. Consequently, a biunique mapping cannot be guaranteed.

Proximity-Based

Unlike the other techniques previously discussed, proximity-based techniques sense the presence of an object, like a NOI, close to a certain point, using a physical phenomenon of limited range. These physical phenomenons involve detecting physical contact (i.e. pressure and temperature sensors), using ID security systems (i.e. computer logins or telephone records), or monitoring access point associations [12, 21]. Regardless of the type of physical phenomenon implemented, the critical component of proximity-based techniques is in gathering and identifying the object's *signature*, or fingerprint. Thus, proximity-based techniques are commonly referred to as either signature-based techniques or *fingerprinting*. For this report, we consider using RSSI-based fingerprinting.

The unique characteristic that differentiates a location, or a fingerprint, may consist of RSS measurements, signal-to-interference ratios (SIR), time delays, or even the channel impulse responses (CIR) [23, 24, 25, 26]. Furthermore, a location within an environment is often expressed as a set of rectangular grid points. Each location's unique fingerprint is first gathered in the *offline* phase. By way of site-surveying, the RSS (or any desired measurement) from multiple nodes, or access points (AP), is gathered at each grid point. The collective fingerprints form a database called a *radio map*, $\left[FP_1', FP_2', \ldots, FP_n'\right]$ for n AP's.

The actual fingerprint of the location is the vector of *mean* RSS values, $[FP_1, FP_2, \ldots, FP_n]$ for n AP's [27].

The real-time phase of fingerprinting is called the *online* phase. In this phase, the node will measure the RSS from different AP's to create a sample fingerprint vector. This sample fingerprint is often centrally processed by the node itself or by a separate entity like a WLAN server. To locate the NOI on the grid, the computing entity compares the sample fingerprint to the radio map and reports the estimated position. Location estimation can be achieved by computing the Euclidean distance, applying Bayesian modeling, or using statistical learning [28, 29, 30].

Consider the the measured fingerprint tuple $[FP_1, FP_2, \ldots, FP_n]$ obtained by the NOI. The Euclidean distance (L_2 distance) between the measured fingerprint and database entries $\left[FP_1^{'}, FP_2^{'}, \ldots, FP_n^{'}\right]$ in the radio map are given by:

$$L_2 = \sqrt{\sum_{i=1}^{n} \left|FP_i - FP_i^{'}\right|^2} \tag{2.13}$$

The estimated position is chosen as the location corresponding to the fingerprint with the minimum distance to the measured fingerprint, or

$$(\hat{x}, \hat{y}) = \min_{FP_i} \sqrt{\sum_{i=1}^{n} \left|FP_i - FP_i^{'}\right|^2} \tag{2.14}$$

In comparison, the Bayesian modeling and statistical learning use stochastics to predict and correct where a NOI is located, a similar approach is used in robot localization. The observations, or the measured RSS of the AP's at different time instances, yield an estimate of the NOI's position. These observations are then used to estimate the NOI's position by correlating them with the radio map. This result then becomes the posteriori position.

A probabilistic approach in estimating the NOI's position relies heavily in using Bayes theorem. Consider the events A and B. Bayes' theorem states that the conditional probability of A given that B is known, $P\left(A \middle| B\right)$, can be determined by

$$P\left(A \middle| B\right) = \frac{P\left(B \middle| A\right) P(A)}{P(B)}, \quad P(B) \neq 0 \tag{2.15}$$

where $P\left(B \middle| A\right)$ is the likelihood, $P(A)$ is the prior probability, and $P(B)$ is the normalizing constant. For fingerprinting, let $P(i)$ denote the probability of receiving measurement i, $P(l)$ be NOI's perceived position at location l, and $P\left(i \middle| l\right)$ is the probability of getting measurement i while at position l. Then the conditional probability of that a NOI is location l given measurement i is:

$$P\left(l \middle| i\right) = \frac{P\left(i \middle| l\right) P(l)}{P(i)}, \quad P(i) \neq 0 \tag{2.16}$$

The general idea in Equation (2.16) forms the basis of the iterative two-step (Bayesian) localization process. The first step is called SEE, where the NOI's perceived location and measurements are mapped to a refined belief state, i.e. determines $P(l|\, i)$. With the calculated posterior probability, the ACT step can occur. ACT will map the posterior probability, or belief state, to a new posterior probability. Mathematically, ACT can be written as:

$$P\left(l_t|\, l_{t-1}\right) = \int P\left(l_t|\, l'_{t-1}\right) P(l'_{t-1})\, dl'_{t-1} \tag{2.17}$$

where l_t is the current location and l_{t-1} is the previous location. Note that the ACT step sums all the possible combinations in which the NOI may have reached l.

When deploying proximity-based location systems, the offline process of establishing the radio map can be very slow and laborious. Another potential disadvantage is that the granularity of the grid size can directly affect the performance of estimating the NOI. Thus, a great challenge in implementing a proximity-based system is how to efficiently collect the fingerprints to construct the radio map without sacrificing system performance.

2.4 Summary

This chapter laid the foundations of position location. A NOI or POI can be localized using either range-based or range-free measurements and techniques. The primary differences between range-based and range-free techniques are the (i) lack specialized hardware for measurements and (ii) the notion that specialized measurements are not available. Figure 2.4 provides a high-level summary of position location. The gathered range-based or range-free measurements are always inputted into a one or a combination of position location techniques. The desired position location technique will output an estimated position, which can then be later optimized by incorporating the NOI's mobility, environmental factors, etc. The topic of optimizing the estimated position will be discussed in a later chapter.

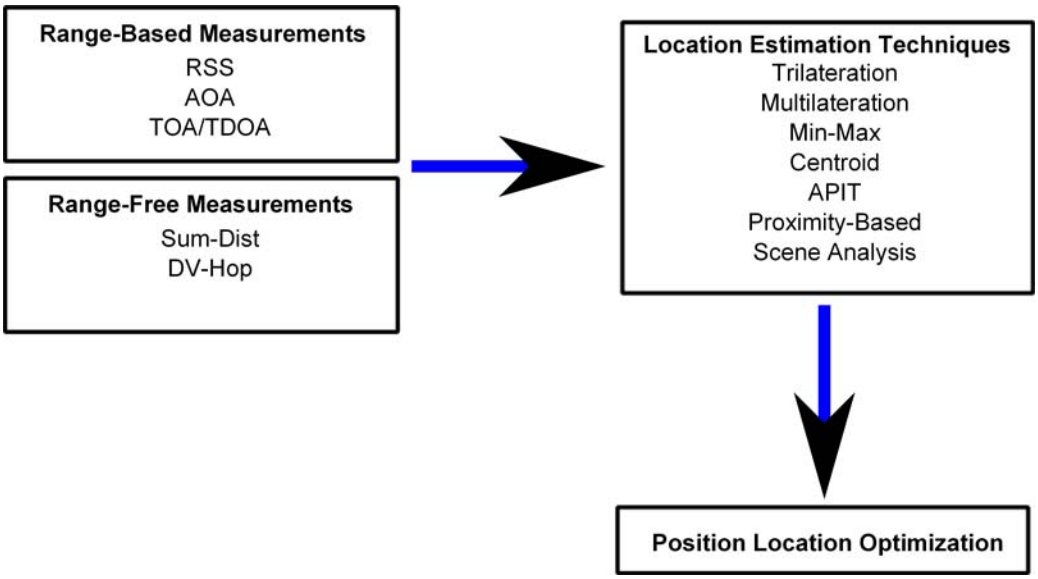

Figure 2.4: Position Location High-Level Overview

Chapter 3

Wireless Distributed Networks

3.1 Overview

The recent advancements in radio technology have ushered the birth of wireless distributed computing (WDC), the concept of collaborative computing in wireless networks, to execute wireless services. The main idea of WDC is to form a distributed computing network from independent radios, which collaboratively work together to accomplish a task. This chapter will discuss the basic technical framework of WDC, especially how WDC differs from traditional distributed computing (DC), and highlight recent results in the area of position location in cognitive radio.

3.2 The Emergence of WDC

In the spirit of DC, the new field of WDC involves the execution of network centric problems with distributed algorithms [31]. Yet the sequential and/or concurrent processes of the distributed algorithms are coordinated through wireless communications. Consequently, the paradigms used in DC are not directly applicable to the dynamic radio environment of wireless networks. One method in meeting the challenges of wireless networks lies in the new technologies of cross layer design and cognitive radio. These driving forces will help accomplish WDC's major goals of reducing per-node and network power, energy, and processing resource requirements; improving security; and enabling novel applications not otherwise possible.

3.2.1 The State of the Art

The ubiquity of commercial smart phones, the increasing popularity and role of commercial and tactical mobile ad-hoc networks (MANETs), and the unparalleled scenarios in modern warfare demand efficient communication networks to support a variety of resource intensive services. For example, the communication and computational tasks associated with tactical wireless sensor networks (WSNs) require strict security and quality of service (QoS) constraints, including power consumption, latency, and communication range, while maintaining stringent resource management requirements. Hence, one radio may not meet all the imposed requirements because of resource constraints or limited functionality. Now if there are multiple resource or functionally limited radios, a WDC network can be formed to overcome individual radio shortcomings to accomplish the mission or provide information services.

WDC marks a cultural shift in how information services are provided in wireless communications. Centralized information services, ideally, would be phased out or augmented with distributed services.

Enabling Technologies

The key technologies of fault tolerant computing, distributed computing, software defined radio (SDR), and cognitive radio (CR) have propelled research in WDC. Their capabilities such as cloud computing with wireless devices flexible link design, and adaptation and autonomous operations enable WDC. For example, recall that early wireless devices had limited access to underutilized processing power and information on peer nodes. Now, radio devices, including smartphones, are used as for communication and sensing/processing information. Collected information can then be shared with other radio nodes and therefore stimulate new applications and enhance current radio capabilities. To communicate with peer nodes, flexible use of wireless resources in meeting QoS needs is accomplishable with CRs and SDRs.

3.2.2 Benefits of WDC

Local information processing demands high resources per node. Furthermore, QoS is not guaranteed if a radio node is fault prone. Thus, the potential benefits of WDC on a collaborative radio network are as follows:

- Lower energy and power consumption per node, which translates to extending total network lifetime;

- Efficient load balancing across collaborating nodes creates better resource demand-supply matching;
- Meeting computational latency requirements of complex processing tasks by harnessing available network computing resources;
- Achieving robust, secure, and fault tolerant execution of information services;
- Simplifying the radio's form factor design;
- and supporting applications that employ location aware service processing and spatial application execution.

Beneficial Applications

WDC is not suitable for all SDR and CR applications. Hence, the types of applications that may benefit from WDC fall into one of these classes:

- Distributed information gathering, real-time processing and dissemination;
- Generation of complex waveforms that extend the range as well as system lifetime;
- Information sharing among units;
- Real-time process control using wireless sensors and actuators.

Examples of distributed applications include wireless grid computing; pattern recognition; channel source coding and decoding [32]; database searches; synthetic aperture radar (SAR); jamming; collaborative signal classification; and position location.

3.3 Modeling

This section presents the fundamentals of WDC. First discussed are the various aspects of a WDC model. Then the focus shifts to quantifying and analyzing performance objectives.

3.3.1 Modeling

A WDC model must incorporate aspects of traditional state-machine composition models found in DC [36] and graph theory used in modeling multi-layer behavior in wireless networks [31]. We now will discuss how elements of DC and wireless networks modeling are applied.

To model general communication, a directed graph is used, where $N_{in}(x)$ and $N_{out}(x)$ denote the in-neighbor and out-neighbors of vertex x. Then the point-to-point channel is modeled as a reliable first-in-first-out (FIFO) channel, reliable ordering channel, and erroneous channel [36]. Note that these channel models remove physical layer details such the channel characteristics and may include randomly distributed physical and link layer parameters.

Intra-node processes are subdivided into a computing process and communication process. In the i^{th} note, the computing process is modeled in terms of its state, denoted $state_i(r)$ at round r, which encodes input/output variables [36]. The stability and correctness of any distributed algorithm depends on the initial states and state transitions. Furthermore, each round r will have message generations and state transitions occurring. Execution trace, or synchronous executions, describe a sequence of events, which can be appropriately modeled as random processes with associated random variables representing events and state transitions by probabilistic model like Markov chains [32]. Therefore, each state has a probability distribution for each r and the process's final state can be described probabilistically for all inputs and failure patterns [36].

3.3.2 Performance Metrics

A WDCN consists of heterogeneous nodes of differing capabilities and resource availability. Moreover, each node individually has a power, communication, and computation subsystem. This heterogeneity across various levels of the network consequently creates a unique challenge to quantify and analyze network system performance. Recall that WDC seeks to minimize both computational and communication latency and power consumption subject to QoS requirements, so novel metrics were developed with this in mind. Metrics associated with QoS are first explained with particular emphasis on time metrics. Lastly, system costs of communication and energy complexity are examined.

One important aspect of QoS is to minimize total execution delay, or time complexity, which is the elapsed time between the start and end of a computation by a WDCN node. We note that the node initiating the distributed computation is not necessarily the same node that finalizes it. This is modeled, in general, as a random variable. Another QoS metric is communication delay, which is the total delay involved in transmission, queuing, and message processing delays. Each of these delays are individually modeled as a random variable dependent on factors such as resource availability and traffic. In comparison, DC systems assume finite and deterministic upper bound delays in ideal conditions of synchronization and fault occurrence [33].

System costs are broken into communication complexity costs and energy complexity costs. We first look at communication complexity costs. The amount of communication activity, i.e. transmission, reception, relaying, resulting while executing a distributed algorithm is communication complexity, or message cost. It can be measured in either messages for higher layer performance analysis or in bits for physical layer (power/energy) evaluations. The following are metrics related to communication complexity [33]: mean complexity per node and mean transmission load per link. Note that communication complexity directly impacts delay and resource utilization, which is important in analyzing network congestion in WDCN. Thus, DC computation complexity metrics are not suitable because they do not consider dynamics of wireless links.

Encompassing both communication and time complexities and energy consumption defines the idea energy complexity system costs. Traditional wireless metrics of energy per raw data bit, throughput, and network energy (power) savings [34] can be applied to WDC.

3.4 WDC System Design

The radio environment greatly influences how the WDCN is modeled, the design tradeoffs, WDC fundamental limitations, and network scalability. Since the design tradeoffs generally entail application and communication layer interactions and inter-system influence of communication and power subsystems, WDC uses a cross-layer and cross sub-system (communication-computation-power) approach to system design. In this section, we look at the three major elements in WDC system design. The elements are communication subsystem design, synchronization, and WDC system control.

3.4.1 Communication Subsystem Design

As explained in an earlier section, message passing occurs during algorithm execution. Message passing also occurs between distributed process during algorithm and WDC control such as synchronization and resource allocation. Therefore, messaging is critical in a WDC. To attain a robust communication subsystem, these three aspects must be considered: scatter and gather, routing and relaying, and/or buffering. The scatter and gather approach requires an efficient MAC capable of minimizing the link interference and can schedule message transmission in synchronization with the process execution. Routing and relaying looks into WDC routing decisions, resource allocation, and efficient means in sharing communication burdens across nodes. Lastly, buffering can control message losses and delays in WDC if bottleneck is formed due to the limited and variable data rates of the wireless link.

3.4.2 Synchronization

In the area of distributed algorithms, synchronization is defined as the establishment of a temporal relationship between events in occurring in different processes [35]. The first element in synchronization is process synchronization. Consider the scenario where event $a_{q,r+1}$ in round $r+1$ of process Q must occur after event $a_{p,r}$ in process P. In order for events $a_{q,r+1}$ and $a_{p,r}$ to be synchronized, P must send a message to Q indicating that $a_{p,r}$ is completed and triggering $a_{q,r+1}$. The sequence of events in P must be executed such that the message indicating $a_{p,r}$ near completion prior to Q beginning $a_{q,r+1}$. The problem is termed the fundamental synchronization problem [35]. Figure 3.1 graphically depicts the fundamental synchronization problem. The τ term indicates the difference in time reference for P and Q and is modeled as a random variable.

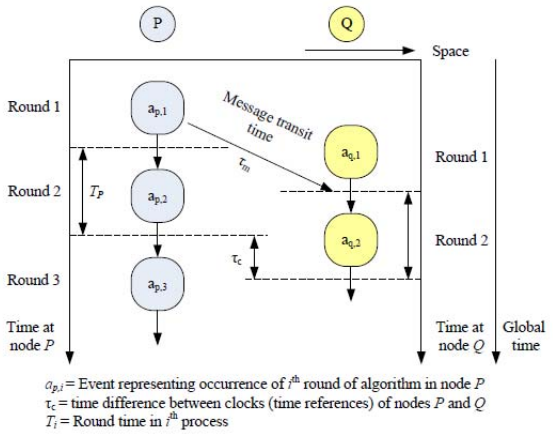

Figure 3.1: Fundamental synchronization problem from [31]

Clock synchronization, the need to synchronize the local clock on each process to maintain an accurate local notion of the global clock, is the second element of synchronization. Synchronized local clocks are used to time stamp events, to re-sync the entire WDCN, and fault tolerance [35]. Intra-node processes also required synchronization in optimizing communication buffering. Note that WDC message transit time is not assumed to be deterministic, which greatly affects determining message transit time bounds for synchronization.

3.4.3 WDC System Control

System control involves coordinating cooperation between all processes, computational task segmentation, and adaptive space-time allocation of sub-tasks assigned to nodes. Though distributed control solutions are very efficient with concurrent control processing and less

communication complexity [35], system control can be centralized, distributed, or a combination of both. This subsection highlights the five key aspects of system control: leader election, topology control, mutual exclusion and resource allocation, scheduling and load balancing, and network information management.

The leader process is a key component in DC algorithms, where this centralized process may perform special functions in addition to the distributed computations. In DC, the *leader* is chosen deterministically. Yet the WDC leader process is subject to the dynamics of the network topology and computation-communication costs incurred by the *leader*. In distributed control, multiple leaders may be utilized, and control functions must be adaptively allocated.

Topology control involves clustering a subset of collaborative nodes based on their link quality to construct a topology. The topology is chosen such that its objectives (e.g. power, communication complexity, connectivity, network reliability, etc.) are optimized. In WDC, connectivity is dependent on several factors: link quality, transmit power, receiver sensitivity, and resource availability.

In the case where the allocation of a single, indivisible non-shareable resource that supports only one user at a time among n concurrent entities, $U_1, \ldots, U_n$, is the mutual exclusion problem. This problem is more prominent in WDC since resources are shared opportunistically [31]. Hence, the resource manager plays an important role in WDC in allocating access of non-shareable resources and communicating with other manager processes. This dynamic managerial process of resources, which consists of maintaining a queue of indicies of user processes to a specific resource, is called resource allocation. For WDC, fair access to queues and stacking of resource queues are the most notable challenges of resource allocation. Since transit delays are random from link to link, some processes may not have fair access to a resource queue in WDCN and face contention with intermediate resources.

The last elements in WDC system control are scheduling and load balancing. Scheduling has three stages: task segmentation and identification of concurrency, allocation of sub-tasks to nodes, and temporal scheduling of sub-tasks within each node [31]. Since WDC employs cross-layer and cross subsystem design, task allocation will not necessarily be uniform among nodes. Then task allocation, in turn, searches for an optimal scheduling strategy over a multi-dimensional space. Figure 3.2 depicts this general WDC task allocation problem. The task execution tree consists of sub-task i's resource requirements (C_i) and communication complexity and tolerable latency between processes executing it (R_i). The network graph shows an abstraction of the nodes and their respective links. Thus, the scheduling and task allocation optimization problem hinted earlier is formulated to balance the workload among nodes subject to the resource constraints (i.e. channel conditions, current load of nodes, and computing and communication resource availability) and cost functions that quantify both communication and computation aspects (i.e. throughput, reliability, concurrency, and complexity). Additional constraints to the scheduling optimization problem include the dynamics of network connectivity, resource availability, and time-variant node availability. We also note that the task allocation optimization problem is further subject to synchronization,

fault tolerance, and security on nodes.

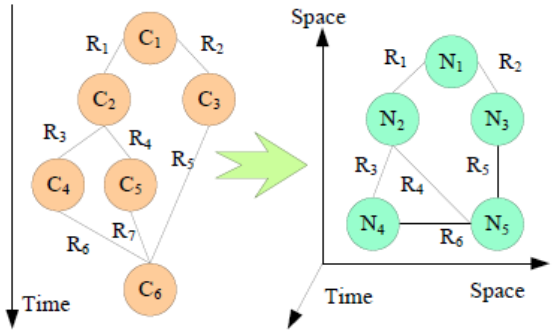

Figure 3.2: The WDC task allocation problem involves mapping the tasks execution tree to a multi-dimensional network graph from [31].

The final aspect of WDC system control is to determine how much information is shared to each node about the WDC environment, or network information management. The associated challenge is how to effectively sample and quantize the dynamic global radio environment state and disseminate this state information to leaders to perform the critical applications of stability analysis, failure recovery, and algorithm correctness [35]. There is no current WDC research investigating this sub-problem of WDC.

3.5 Location Awareness for the WDC Paradigm

The previous sections gave an overview of the new research area of WDC and highlighted how it contrasts with DC. So how does position location fit in the context of WDC? The goal of this section is to answer this question and underscore the importance of position location in the WDC paradigm.

In CR, awareness is very important, especially the device's location so that it may sense the environment and dynamically adapt to the changing conditions [37]. This task is known as *location awareness* [38, 39]. Hence, location information is exploited to improve the overall wireless communication system. Applications of location awareness are found in location-based services (i.e. bill routing, social media, etc.), environment characterization, network optimizations, and transceiver optimization.

For WDC, position location information of nodes and network resources, or simply location awareness, is one element of the WDC environment state. Consequently, optimization of network resources and communication complexity to meet certain QoS requirements is indirectly dependent on the physical position of nodes. Another location dependent aspect of WDC system control is topology and routing control, especially in determining feasible and

realizable network topologies. Note that position information can be determined using the techniques discussed in Chapter 2. Regardless of the positioning technique(s) used, the key issue in location awareness is how to adapt in order to achieve interoperability with other technologies like Bluetooth, CDMA, WiMAX, and satellite WDC nodes may possess.

In order to leverage location awareness, the WDC system, with the help of CR, has to assess or classify the radio environment [40]. Yarkan and Arslan in [40] recommend using a database outlying appropriate statistical propagation models for each environment. Once the propagation characteristics of each link is determined and disseminated, individual nodes can adaptively improve their resource allocation, scheduling, and load balancing. The task of developing the propagation database is tedious, laborious, and should be completed prior to deploying a WDCN. In some instances, an *a priori* environment database may be infeasible or pose security risks in tactical scenarios.

3.6 Summary

This chapter gave an overview of the current developments in WDC. The novel paradigm is a significant departure from centralized to decentralized, or distributed, information services. Unlike in DC, the wireless radio environment greatly influences how the WDCN is modeled, the design tradeoffs, WDC fundamental limitations, and network scalability. Because the wireless radio environment is inherently dependent on the position of a transceivers, location awareness is a key component in the WDC environment state.

Chapter 4

Distributed Positioning Algorithms

4.1 Introduction

Building on the positioning techniques discussed in Chapter 2, this chapter forms the foundations of distributed positioning algorithms for WDC systems. In particular, we focus on indoor-based WDC systems and subsequently the indoor positioning problem. Section 4.2 briefly reviews the fundamentals of the indoor wireless channel. Then Section 4.3 clarifies the indoor positioning problem and potential solutions to it. Thirdly, the associated distributed algorithms used to solve our indoor localization problem are explained in Section 4.4. Lastly, Section 4.5 looks into the position location performance metrics and their applicability to our research.

4.2 Indoor Wireless Channel

In comparison to RF outdoors, the indoor wireless channel cannot be solely captured with path loss exponents. This unpredictability of RF indoors is due to three major attenuation factors: attenuation due to distance, losses through walls and floors, and severe multipath propagation. This section will briefly explain the aforementioned attenuation factors and other challenges faced in the indoor wireless environment.

Firstly, signal attenuation over distance occurs when the mean received signal power attenuates as a function of the distance from the transmitter. This type of attenuation is commonly seen in the form of the free-space path loss equation,

$$PL(dB) = 20\log_{10}(d) + 20\log_{10}(f) + 20\log_{10}\left(\frac{4\pi}{c}\right) \tag{4.1}$$

where f is the signal's frequency in Hertz, c is the speed of light ($3x10^8$ m/s), and d is the

distance from the transmitter to the receiver in meters. Equation (4.1) assumes that the signal power spreads out over the surface area of an increasing sphere as d increases. In addition to free-space path loss effects, signals may experience ground wave losses. Though not nearly as prominent indoors, ground wave loss can occur when signals propagate through hallways.

By far the most significant attenuation factor in indoor radio environments is multipath. Recall from Chapter 2 that multipath is caused by the reception of multiple copies of a transmitted signal. Hence, multipath leads to RSS variations over frequency and antenna location.

The Saleh and Valenzuela indoor statistical model is a popular channel model, which characterizes the indoor wireless multipath channel [41]. Based on their measured channel impulse responses, they present a simple multipath model assuming that the multipath components arrive discretely in clusters. Each multipath component within a cluster form a Poisson arrival process with different rates λ. The channel impulse response will have the following mathematical form:

$$h(t) = \sum_{l=0}^{L} \sum_{k=0}^{K} \beta_{kl} e^{j\phi_{kl}} \delta\left(t - T_l - \tau_{kl}\right) \tag{4.2}$$

where L and K are the number of clusters and arrivals within each cluster, respectively and T_l is the arrival of the l^{th} cluster. Each multipath arrival is characterized by a delay within the cluster τ_{kl}, amplitude β_{kl}, and phase ϕ_{kl}. Note that β_{kl} are independent Rayleigh random variables and ϕ_{kl} is uniformly distributed between $[0, 2\pi]$.

Furthermore, indoor delay spreads, or the degree of time variation, are smaller (nanoseconds versus microseconds) than those experienced in outdoor measurements. RF indoors do not, in general, experience Doppler spreading, which suggests frequency spreading should not occur. Lesser sources of signal degradation include temperature and humidity variations, relocating of furniture and/or equipment, and even the presence or mobility of human beings.

Another notable difference between the indoor and outdoor radio environment is that indoor frequency bands are generally higher than cellular case. Thus, changes to a higher frequency affects propagation negatively since signal strength is inversely proportional to at least the square of the frequency.

4.3 Indoor Positioning

Consider the scenario where a cellular base station (BS) desires to localize a mobile station (MS). Abstracting the cellular technology used by the BS and MS, there are general two ways to localize the MS: by using GPS or enhancing the existing communications infrastructure. As highlighted in Chapter 1, GPS is a reliable outdoor positioning system, providing a positioning accuracy of 7 to 10 meters with highly capable GPS receivers. Its positioning

accuracy can be further improved to 3 to 5 meters with differential corrections. Yet GPS is not suitable for every situation because of the lack of coverage and the associated costs to erect GPS infrastructure. In the case of localizing an user, which may be a human or non-human, in an indoor environment, GPS is clearly not applicable as a positioning system. Therefore, it appears more economical to enhance the existing communications infrastructure.

This section aims to elaborate on the problem of indoor positioning and the related challenges. Moreover, we will discuss solutions to the indoor positioning problem in the context of WDC systems. Lastly, we describe a position location for WDC systems.

4.3.1 Problem Formulation

The indoor localization problem for the WDC system we will focus for the remainder of the report is to localize and monitor the movements of a human user in an indoor environment. More generally, the localization problem is to find the relative position of a node transmitting indoors using a wireless sensor network. Note that this is the reverse position location problem of localizing a receiver with transmitters.

4.3.2 Localization Approaches

There are several approaches to solving our indoor localization problem. Recall from Figure 2.4 the different position location techniques. Of the ones mentioned in this report, the three most commonly used techniques are triangulation, determining the AoA, or employing proximity based schemes. Yet the harsh multipath environment makes triangulation or finding AoA difficult and yield erroneous results [42, 43, 44]. Thus, proximity based schemes, especially fingerprinting appear attractive.

As mentioned in this section's introduction, enhancing the existing communications infrastructure is a cost effective approach in tackling indoor localization. However, triangulation and AoA both require highly sensitive receivers, which may become increasing costly to upgrade radio equipment covering a large geographic area. In comparison, fingerprinting is most feasible because it is capable of using the existing communication framework. Chapter 6 will delve more in the types of fingerprinting-based positioning systems.

4.4 Distributed Localization

In the context of wireless sensor networks, distributed localization refers to how sensor nodes localize themselves within the network. These distributed algorithms are generally employed

in large-scale sensor networks of 100+ nodes [13]. The algorithms themselves fall into one of these categories [45]:

- Network multilateration - Each node estimates its range to the nearest anchor nodes. When each sensor has multiple range estimates to known positions, multilateration is calculated locally.
- Successive refinement - Algorithms of this type will find the optimum of a global cost function, e.g. least squares (LS), weighted LS (WLS) [46], or maximum likelihood (ML). The node's estimated location is then transmitted to neighbors, where neighbors must refine their position until convergence.

Regardless of the type of distributed localization algorithm employed, all undergo a generic, three-phase approach [13]:

1. *Distance Phase* - Determine the distances between unknown nodes and anchor nodes.
2. *Positioning Phase* - Derive node position from anchor distances.
3. *Refinement Phase* - Successively refine the node positions using aggregate distances to and positions of neighboring nodes.

In the Distance Phase, nodes will share information to collectively ascertain the distances between their unknown position and anchors. The estimated distances are used to calculate an initial position the Positioning Phase. Phase 2 applies one of the positioning techniques shown in Figure 2.4. The last phase of refinement is optional, but highly recommended since the estimated position in phase 2 may be not accurate.

For a WDCN, distributed localization encompasses two concepts: (1) self-localization of WDC nodes and (2) localizing an target in a distributed manner. To distinguish between the two concepts, we will use the term *start-up algorithm* to denote self-localization of nodes. Then *distributed localization*, or *distributed target localization*, will be used to describe the application of position location, i.e. finding a POI. More information about the algorithms used for distributed localization will be discussed in the upcoming subsection.

Though WDC systems, generally, are not large scale, the self-localization process previously described can be adapted for the start-up algorithm in WDCN's. The Distance and Position Phases would occur in the network's offline phase, where the WDC system's power consumption model is developed. Then the third phase could occur in the online phase, pending the urgency of the WDCN's application. The outlined process best describes the high-level overview of the methodology used to localize WDC assets for two main reasons. Because it is application independent, we can use any type of position measurement or position technique best suited for the scenario in Figure 2.4. More importantly, the process accommodates both the outdoor and indoor wireless environments.

To further emphasize the simplicity and adaptability of the three-phase start-up algorithm, we consider the following WDCN with one leader, or master node, a anchor nodes, and n sensor nodes. For simplicity, we assume the master node is also an anchor node. All a anchor nodes "know" their positions *a priori*. The method as to how their position is determined is not of concern; it may have been resolved using GPS or manually programmed by an external source. Nonetheless, the exact locations of the anchor nodes populate the following matrix,

$$A = \begin{bmatrix} B_1(x,y) \\ B_2(x,y) \\ \vdots \\ B_i(x,y) \end{bmatrix}, \quad i = (1, 2, \ldots, a)$$

where B_i is the 2-D coordinate of an anchor node. We will also assume that all network nodes are synchronized to a global network clock.

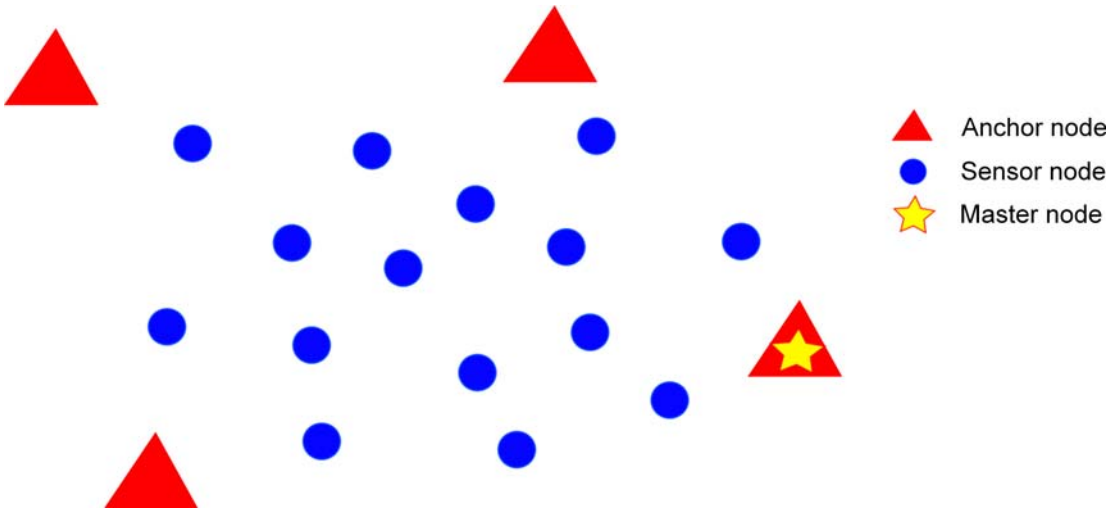

Figure 4.1: Sample WDCN

In the Distance Phase, the anchor nodes broadcast beacons to the n sensor nodes. Each sensor node builds a range vector $\mathbf{b}$ from, from example, TOA measurements. The range vector $\mathbf{b}$ consists of the sensor node's range estimate r_i to the i^{th} anchor node,

$$\mathbf{b} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_i \end{bmatrix}, \quad i = (1, 2, \ldots, a)$$

Simultaneously, to facilitate the second phase, anchor nodes also pre-calculate the distances amongst themselves to form the distance vector $\mathbf{d}_p$. Define d_p as

$$\mathbf{d}_p = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_i \end{bmatrix}, \quad i = (1, 2, \ldots, a)$$

where d_i is the distance between the master node and the a^{th} anchor node. The second phase involves the sensors estimating their position. But beforehand, the master node broadcasts the matrix A and the anchor distance vector $\mathbf{b}$. Now the sensor node can estimate their position as

$$(\hat{x}, \hat{y}) = \hat{\mathbf{x}} = \frac{A_p}{2}\left(r_1^2 - \mathbf{r}^2 + \mathbf{d}_p\right), \quad A_p = \left(A^T A\right)^{-1} A^T, \quad \mathbf{d}_p = d^2$$

Note that the sensor's estimated position is in fact calculated in a distributed least squares sense [47]. For this example, the refinement phase is not used. The entire sensor node process is summarized in Algorithm 4.1.

Algorithm 4.1 Example Start-Up Algorithm for a Sensor

Phase 1 - Distance Phase
while number of beacons received $\leq a$, **do**
  calculate r_i
  $\mathbf{b} \leftarrow r_i$
  if number of beacons $= a$, **then**
    break
  end if
end while
Phase 2 - Position Phase
if A and d_p is received, **then**
  $A_p \leftarrow \left(A^T A\right)^{-1} A^T$
  $\mathbf{d}_p \leftarrow d^2$
else
  wait
end if
$\hat{\mathbf{x}} \leftarrow \frac{A_p}{2}\left(r_1^2 - \mathbf{r}^2 + \mathbf{d}_p\right)$

What we have shown is a simple example of a WDCN start-up algorithm. Now suppose the anchor and sensor nodes have GPS capabilities, then Phase 1 and Phase 2 collapse into a single phase. In this case, the Refinement Phase would significantly increase the node's location accuracy by taking differential GPS measurements.

4.4.1 Distributed Target Localization Algorithms

This subsection looks into the second notion of distributed localization for a WDCN, distributed target localization. We first look at the general case of distributing position location calculations in the outdoor environment. Then we specialize our findings to the indoor localization problem mentioned earlier in the chapter. Note that the success of target localization is dependent on the completion (and success) of the start-up algorithm. Poor sensor node location estimates immediately translate to poor estimates of where the target is. Thus, we assume for the remainder of the report that the sensor nodes' locations are known.

Position location applications such as target localization and tracking can be computationally expensive for a resource-strained radio or sensor node. Particularly, the node may lack sufficient memory in searching and calculating large geographic databases. Depending on the specific application requiring position information, the node may also not have sufficient battery power to securely transmit and process position data. To improve overall temporal granularity and achieve the minimum QoS, parallel processing or distributed computing can be applied. Another important benefit of distributedly computing positions prevents single points of failures. Although there may exist differences between parallel and distributed computing, we will use them interchangeably for the rest of the report.

Parallel or distributed processing systems operate on the principle that large problems (and subsequently large calculations) can be subdivided into smaller problems. They leverage several processors, either close or far apart, to jointly execute one or a series of computational tasks. In the case of position location applications, the computational tasks can include multilateration calculations or even fingerprint database matching. It is this highly repetitive nature of position location calculations that makes it an ideal application for WDC.

Positioning algorithms, in general, can be categorized as calculating a precise position or calculating an approximated position. Precise positioning algorithms are based on solving a linear system of equations such as multilateration or triangulation. In comparison, approximate positioning algorithms often involve non-linear equations, which may not converge to a solution.

In parallelizing or distributing the localization calculations, there are two major approaches:

1. Transfer the more complex aspects of the calculations, such as matrix multiplication or matrix inversion, to a master node,
2. Or divide the calculations amongst a subset of nodes.

The first approach will leave the simplest calculations to the resource constrained nodes. An example of this is the algorithm in the previous section. In comparison, the second approach subdivides or parallelizes the calculations by dividing the geographical region in areas of responsibilities (AOR's). This is the approach we will explore more and later use in implementation.

In outdoor positioning systems, the most common calculations involve a multilateration, which, recall, is based on a system of equations. To solve the system of equations, the least squares (LS) approach is frequently used. LS, in its simplest case, is deterministic. If appropriate probabilistic assumptions about underlying error distributions are assumed, LS optimization becomes the maximum-likelihood (ML) estimate of the parameters. The general LS optimization problem simply minimizes the sum of the squares of the errors and is written as,

$$\min_x \|Ax - b\|^2 \tag{4.3}$$

It can be solved directly and numerically using singular value decomposition (SVD), QR decomposition, or normal equations. The iterative approaches to solve (4.3) are applying Gauss-Sidel elimination, Richardson, and the Jacobi. The fastest, least accurate algorithm to solve (4.3) is to use normal equations, while the most computationally expensive and slowest algorithm is SVD.

Consider the general LS problem for an outdoor positioning system, we can column-wise partition matrix $A \in \mathbb{R}^{mxn}$ from (2.5) as follows:

$$A = [A_1 \quad A_2], \quad A_1 \in \mathbb{R}^{mxp}, \quad A_2 \in \mathbb{R}^{nxp}$$

Then we can rewrite Equation (2.5) as

$$[A_1 \quad A_2]\,\mathbf{x} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{4.4}$$

The parallel computation becomes:

$$x_1 = \left(\left(I - B\left(B^T B\right)^{-1} B^T\right) A\right)^+ \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad x_2 = \left(\left(I - A\left(A^T A\right)^{-1} A^T\right) B\right)^+ \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{4.5}$$

We note that Equation (4.4) requires that $[A_1 \quad A_2]$ have full rank. If information is known about A and b, we could rewrite the LS optimization problem as

$$\min_x \|W(Ax - b)\|^2$$

where W is usually the covariance matrix. But because we also seek to reduce the communication overhead between nodes, we could not use this formulation of the LS problem because it requires iterative communication and updates from nodes. Furthermore, Equation (4.4) does not scale well to more than two nodes. Instead, we focus on the second approach where nodes are assigned AOR's.

We have shown that trying to parallelize solving the general LS problem will not minimize the communication power between nodes because of its iterative nature. So we shift focus to the second approach of dividing the calculations to a subset of nodes. The general idea of the second approach is to have nodes running concurrent processes and then later aggregating the information to generate a consensus.

To better illustrate this concept, please refer to Figure 4.2. This is the same WDCN in Figure 4.1 except that we have now delineated AOR's for each anchor node. The anchor nodes are in charge of aggregating and collecting position information from the sensor nodes for their assigned area. Once all position information is fused, the anchor nodes relay the data to the master node via a broadcast or mulit-hop routing. This hierarchy of nodes completely removes most computational dependency on the sensor nodes and shifts it to the anchor nodes [48]. Now suppose a mobile node enters AOR 1. The sensors in the area will detect the presence and report it to closest anchor node. However, the closest anchor node cannot simultaneously support receiving/filtering sensor reports and localize the mobile node without compromise. It, instead, sends a service request to distribute the localization calculations to anchor nodes not currently reporting the mobile node's presence. Once the service request is determined feasible and granted, the anchor node of AOR 1 will compress the gathered measurements and securely broadcast it to the other anchor nodes. The anchor nodes in AOR 2, AOR 3, and AOR 4 will then concurrently process the measurements and report their calculations to the anchor node of AOR 1.

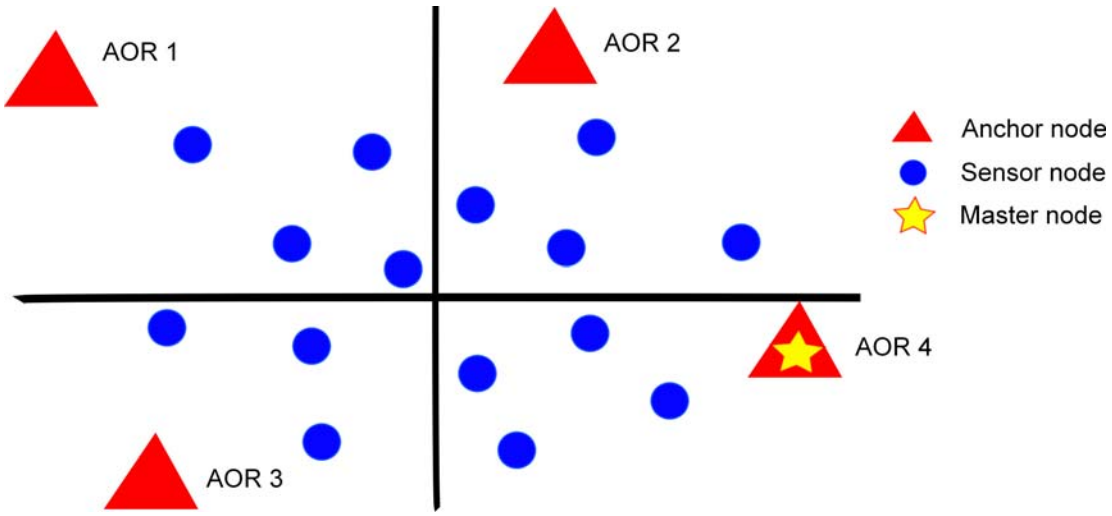

Figure 4.2: Sample WDCN with assigned AOR's

The scenario described above is a classic example of the single-client, multiple-servers arrangement found in DC and WDC. In this type of arrangement, it is important to minimize the time taken to compute the individual tasks, or minimizing the makespan. The minimum makespan is a scheduling problem that is described as:

> Given a set of J of m jobs and a set of N of n nodes, the processing time for a job $j \in J$ on node $i \in N$ is $p_{ij} \in \mathbb{Z}^+$. Then we must find an assignment of the jobs J to the nodes N such that the makespan, or the completion time, is minimized.

Its integer program formulation is:

$$\begin{aligned} \text{minimize} \quad & t \\ \text{subject to} \quad & \sum_{i \in N} x_{ij} = 1, \quad j \in J \\ & \sum_{j \in J} \quad x_{ij} p_{ij} \leq t, \quad i \in N \\ & x_{ij} \in \{0, 1\}, \quad i \in N, \, j \in J \end{aligned} \tag{4.6}$$

Here, the first constraint ensures that all nodes have a fraction of workload. The second constraint establishes an upper bound of t on the processing time for each node. For a WDCN, minimizing the makespan implies synchronization of the subtasked nodes such that decentralized position calculations on separate nodes is significantly lesser when comparing to centralized computations [49]. We must note that this scheduling problem becomes extremely complex when coupled with WDC-related constraints and temporal constraints imposed by the position location application. Without a loss of generality, the algorithms we will discuss next solely focus on minimizing the makespan and do not include WDC-related constraints.

We have shown that parallelizing the LS problem as in wired networks is not ideal for WDCN's because of its highly communicative process which will only further constrain nodes. Instead, we move away from outdoor positioning systems to indoor positioning systems to illuminate the concept of concurrent position calculations. When using this approach, we have assumed that the receivers in the indoor environment are fixed.

As emphasized in an earlier section of this chapter, the indoor wireless environment is extremely prone to multipath. Hence, the fingerprinting technique is most feasible because it is capable of using the existing communication framework. Recall from Chapter 2 the methods of accurately estimating a fingerprint: Euclidean distance (Equation (2.14)) and Bayesian statistics (Equation (2.16)). The Euclidean distance optimization is effectively an unconstrained search of a database. If the database is sufficiently large due to the spatial granularity of fingerprints, this can be computationally taxing on a single node. For Bayesian modeling, the greatest computational burden is in the large matrix multiplications in Equation (2.16) and Equation (2.17). Thus, both of these can optimized to run in concurrent processes. Lastly, we will explore estimating fingerprint positions with distributed neural networks.

Distributed Euclidean Distance

Suppose the fingerprint radio map contains f fingerprints, which correspond to f fingerprint locations. Since we seek to minimize the makespan, the distributed Euclidean distance algorithm (DEDA) partitions the fingerprint database into p partitions and assigns each node a partition for processing, where p_i for $i \in \{1, 2, \ldots, p\}$ is the assigned partition. Each

p_i contains a subset f_j of the f fingerprints. Then we have Algorithm 4.2 and Algorithm 4.3 for an individual processing node and a master node, respectively:

Algorithm 4.2 Distributed Euclidean Distance Algorithm (DEDA)

Initialize $\hat{FP}_i = 0$.
while p_i is assigned and received, **do**
  for all $FP'_j \in f_j$, **do**
    $\hat{FP}_i \leftarrow \sqrt{\sum_{k=1}^{j} \left| FP_k - FP'_k \right|^2}$
  end for
end while
$(\hat{x_j}, \hat{y_j}) \leftarrow \min_{FP_i} \in f_j$.
return $(\hat{x_j}, \hat{y_j})$

Algorithm 4.3 Distributed Euclidean Distance Algorithm (DEDA) for a master node

Broadcast FP measurements to n processing nodes.
$l = 0$
while $l \neq p$ **do**
  if Received $(\hat{x_j}, \hat{y_j})$ **then**
    $l = l + 1$
  end if
end while
$(\hat{x}, \hat{y}) \leftarrow \min_j (\hat{x_j}, \hat{y_j})$
return $(\hat{x}, \hat{y})$

In Algorithm 4.2, the processing node is assigned an AOR dynamically (i.e. its AOR changes for each WDC service request) or statically designated. This AOR effectively limits the search region for the processing node. Note that partitioning is not analogous to constraining the database search. A constrained search will indeed decrease the fingerprint search area, but it will require knowledge of where the user has previously been to optimally confine the algorithm's search. Segregating the database search into concurrent searches does not narrow the fingerprint search. It subdivides the fingerprint database into more manageable search areas by distributing the computational complexity across multiple processing nodes. With an designated AOR, the processing node will calculate the fingerprint position with the minimum Euclidean distance within their AOR. This information is then sent to the master node. In Algorithm 4.3, the master node will compare the answers received from each processing node and only output the fingerprint position with the smallest Euclidean distance.

Distributed Bayesian Modeling

Suppose the fingerprint radio map contains f fingerprints, which correspond to f fingerprint locations. Since we seek to minimize the makespan, the distributed Bayesian modeling algorithm (DBMA) partitions the fingerprint database into p partitions and assigns each node a partition for processing, where p_i for $i \in \{1, 2, \ldots, p\}$ is the assigned partition. Each processing node then computes (2.16) for their respective partition and sends a vector $\mathbf{g_i}$ containing the individual probabilities. Algorithm 4.4 and Algorithm 4.5, featured below, outline DBMA for an individual processing node and a master node, respectively:

Algorithm 4.4 Distributed Bayesian Modeling Algorithm (DBMA)

Initialize $\mathbf{g_i} = \mathbf{0}$.
while p_i is assigned and received, **do**
  for all $FP_j' \in f_j$, **do**
    $\mathbf{g_i}(j) \leftarrow P(j|\{FP_1, FP_2, \ldots, FP_n\})$
  end for
end while
return $\mathbf{g_i}$

Algorithm 4.5 Distributed Bayesian Modeling Algorithm (DBMA) for a master node

Broadcast FP measurements to n processing nodes.
$l = 0$
while $l \neq p$ **do**
  if Received $(\hat{x_j}, \hat{y_j})$ **then**
    $l = l + 1$
  end if
end while
$(\hat{x}, \hat{y}) \leftarrow \max_j \mathbf{g_i}$
return $(\hat{x}, \hat{y})$

The processing node in Algorithm 4.4 will calculate the user's believed position probability for each fingerprint position in their AOR. The posterior probabilities are then sent to the master node for further processing. Upon receiving each processing node's vector of probabilities, the master node in Algorithm 4.5 sums the possible combinations in which the user may have reached a location and reports only the fingerprint position with the maximum probability.

Distributed Neural Networks

A less commonly used method of estimating fingerprint positions employs neural networks [50, 51]. Neural networks are capable of performing complex tasks such as classification,

optimization, control and function approximation. In the case of fingerprinting, neural networks can be viewed as function approximation problem using nonlinear regression, where the inputs of a RSS values, where a is the number of AP's or bases stations used, are mapped onto a set of two output variables representing the 2-D location of the user. The two types of neural networks applicable for fingerprinting are the multilayer perceptron neural networks (MLPNN) and the radial basis function (RBF) networks.

Each type of neural network has an associated learning algorithm. These algorithms adjust the internal network's weights and biases based on the minimization of an error function and defines how the network is trained. MLPNN globally reaches any nonlinear continuous function because of the sigmoid basis functions it uses. In comparison, RBF networks only reach nonlinear continuous functions locally because they cover only small, localized regions. Designing an RBF network is also easier and faster than a MLPNN. We consider applying a distributed generalized regression neural network (GRNN), which is a specialized type of RBF network, and a distributed MLPNN. Both will require a supervised learning phase and a testing phase.

The general neural network architecture is shown in Figure 4.3 for one WDC node. The GRNN and MLPNN architectures consists of twelve inputs corresponding to the twelve RSS values at each sensor, one hidden layer, and an output layer of two neurons corresponding to the (x, y) location of the user (fingerprint location). Like the DEDA and DBMA, each WDC node is assigned an AOR and is trained accordingly.

Since we seek to minimize the makespan, the GRNN and MLPNN first partitions the fingerprint database into p partitions and assigns each node a partition for processing, where p_i for $i \in \{1, 2, \ldots, p\}$ is the assigned partition. Each processing node then computes and analyzes the inputs with their respective neural network sends a vector $\mathbf{h_i}$ containing their estimated position of the user. The master node will then chose the maximum values of (x, y) for each iteration.

4.5 Performance Assessment Metrics

Position location systems, either indoor or outdoor, need performance measures that can describe the accuracy of the location estimate scheme. This section will outline the popular metrics of the Cramer-Rao Bound, Circular Error Probability, and Geometric Dilution of Precision. For a fingerprinting system, these performance metrics are not applicable. Rather, the primary metric used is the average distance error.

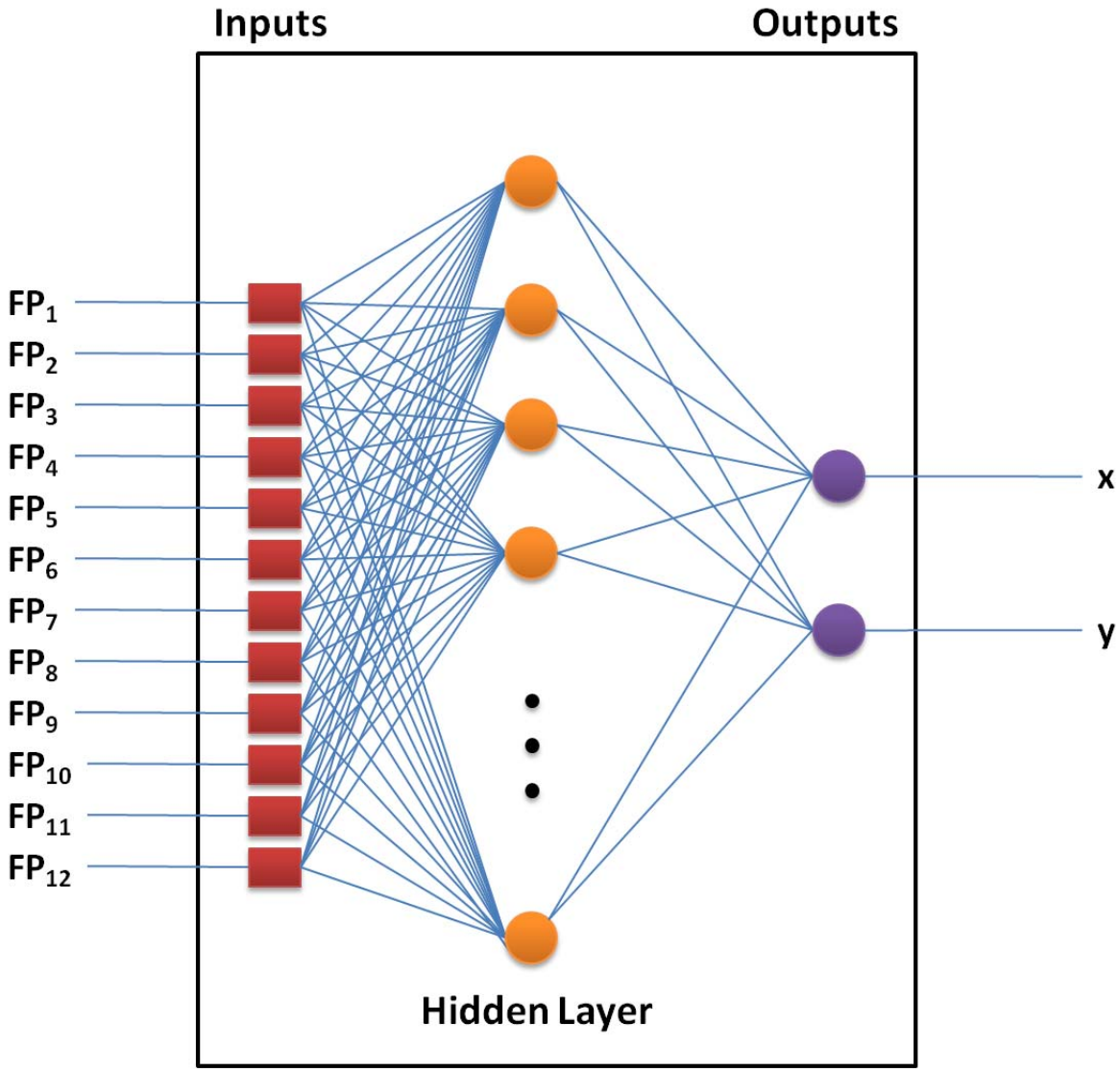

Figure 4.3: Architecture of a neural network

4.5.1 Cramer-Rao Bound

The Cramer-Rao Bound (CRB) provides a theoretical lower limit to the error covariance matrix W of a deterministic parameter, or the estimated position $\hat{\mathbf{q}} = [\hat{x}, \hat{y}]$. Mathematically, W is defined as

$$W = E\left[(\hat{\mathbf{q}} - \mathbf{q})(\hat{\mathbf{q}} - \mathbf{q})^T\right] \geq J^{-1}$$

where J is the Fisher information matrix, defined as

$$J = -E\left\{\frac{\partial}{\partial\theta}\left(\frac{\partial}{\partial\theta}\log p_q(\mathbf{r})^T\right)\right\}$$

and $p_q(\mathbf{r})$ is the joint probability density distribution of observation $\mathbf{r}$. Here $\mathbf{r}$ is the noisy measurement.

If measurement noise is Gaussian with zero mean and has a covariance matrix K, then $\mathbf{r}$ is also Gaussian with covariance matrix K. It's mean vector, though, will be a function of $\mathbf{q}$. The CRB for $\hat{\mathbf{q}}$ is simply [52]:

$$CRB = J^{-1} = \frac{\partial f(\mathbf{q})^T}{\partial \mathbf{q}} K^{-1} \frac{\partial f(\mathbf{q})}{\partial \mathbf{q}}$$

4.5.2 Circular Error Probability

In comparison to the CRB, the circular error of probability (CEP) measures uncertainty in $\hat{\mathbf{q}}$ relative to its mean $E[\hat{\mathbf{q}}]$. It may be derived by solving:

$$\frac{1}{2} = \int\int_R p_{\hat{q}}(\xi)\, d\xi_1 d\xi_2$$

where $p_{\hat{q}}(\xi)$ is the probability density function of a $\hat{\mathbf{q}}$ and $R = \{\xi :| \xi - E[\hat{\mathbf{q}}]\} \leq CEP$ [53]. It can be approximated rather as

$$CEP \approx 0.75\sqrt{E\left[(\hat{\mathbf{q}} - E[\hat{\mathbf{q}}])^T(\hat{\mathbf{q}} - E[\hat{\mathbf{q}}])\right]} = 0.75\sqrt{\lambda_1 + \lambda_2}$$

where λ_1 and λ_2 are the eigenvalues of this estimator covariance matrix

$$E\left[(\hat{\mathbf{q}} - E[\hat{\mathbf{q}}])^T(\hat{\mathbf{q}} - E[\hat{\mathbf{q}}])\right] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_1^2 \end{bmatrix}$$

4.5.3 Geometric Dilution of Precision

The last traditional position performance metric is the geometric dilution of precision (GDOP). It is a measure of the effect of the geometric configuration of the land or satellite references

on the location estimation. Note that GPS is the primary location system that relies on GDOP to ascertain the quality of the estimated position. Mathematically, it is defined as the ratio of the root mean-square position error to the room mean-square ranging error or

$$GDOP = \frac{1}{\sigma_r}\sqrt{E\left[\left(\hat{\mathbf{q}} - E[\hat{\mathbf{q}}]\right)^T\left(\hat{\mathbf{q}} - E[\hat{\mathbf{q}}]\right)\right]}$$

where σ_r is the fundamental ranging error [53].

4.6 Chapter Summary

This chapter first explained the indoor positioning problem and the challenges not seen in the outdoor environment. Then we delved into how to parallelize localization algorithms to minimize the makespan of a distributed task. Most notably, we looked into the optimization algorithms of DEDA, DBMA, and neural networks for fingerprint positioning systems. Both DEDA and DBDM are the chosen distributed positioning algorithms that will be implemented in a WDC system, which is the subject of the next chapter.

Chapter 5

Modeling of the WDC Positioning System

5.1 Introduction

The proceeding chapters have given a thorough overview of position location, WDC, and distributed localization algorithms. This chapter will now leverage the aforementioned topics in modeling a WDC-based fingerprinting positioning system. The literature review in Chapter 2 pointed out that signal strength's attenuation is caused by one of three independent factors: path loss, multipath fading, and shadowing. Therefore, the initial data analysis is extremely important in fingerprinting. A better understanding of the RSS properties in ICTAS can assist in improving our distributed localization algorithms and in deploying the WDC system. First, Section 5.2 will focus on the characteristics and analysis of the RSS measurements taken prior to deploying the system. Then Section 5.3 assesses the proposed distributed localization algorithms from Chapter 4 with experimental data.

5.2 Properties of RSS in ICTAS

Fingerprinting's position location accuracy highly depends on the initial RSS data analysis. This analysis of the RSS measurement is extremely important in understanding the intrinsic features of location-dependent RSS patterns and location fingerprints. Understanding the underlying characteristics of the data will enable us in honing our proposed distributed localization algorithms and improve our system design.

5.2.1 Measurement Campaign

Location

The location chosen for the measurement campaign is Virginia Tech's Institute for Critical Technology and Applied Science (ICTAS) headquarters building (Figure 5.1). All measurements were conducted along the first floor's main corridor. One major item to note is that the floor elevation on the first floor is unique in that it has a slight incline, causing about a one foot difference in height from each ends of the passageway (Figure 5.2). ICTAS' other floors do not have this unusual feature.

Figure 5.1: Virginia Tech's ICTAS Headquarters on Stanger Street

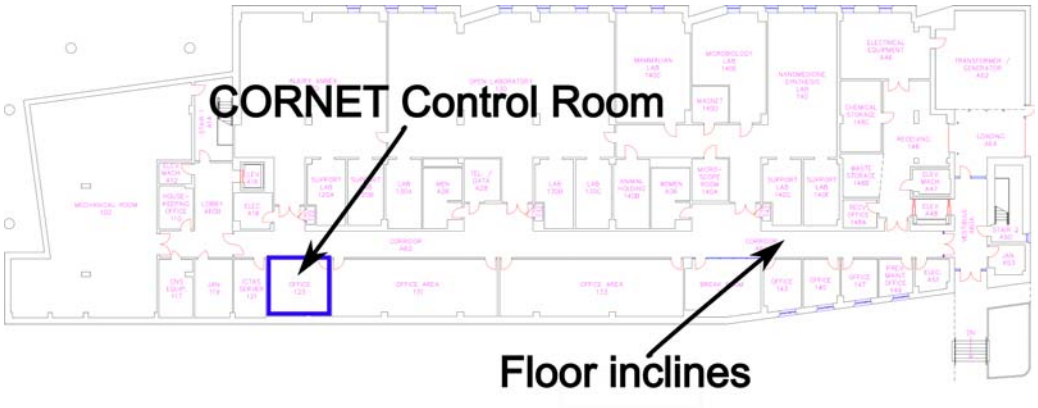

Figure 5.2: ICTAS' First Floor Schematic

Hardware and Software

A Sony VAIO VPCEB laptop computer equipped with a Universal Software Radio Peripheral 2 (USRP2) [54] and a Motorola Talkabout MR350R radio [55] were primarily used in collecting RSS samples from twelve nodes part of Virginia Tech's Cognitive Radio Network Testbed (CORNET) [56] (Table 5.1). CORNET is a collection of high performance servers

and 48 unique, flexible SDR nodes with USRP2 RF front-ends. The 48 nodes are distributed throughout ICTAS, on all four floors. Note that the USRP2 used as a transmitter (see Table 5.1) is a separate device and not connected to CORNET. Also, the frequency band used is the Family Radio Service (FRS) channel 17, or 462.6 MHz. Due to construction work in the adjacent building to ICTAS, operating on FRS channel 1, 462.5625 MHz, could not be actively used.

Table 5.1: Fingerprinting Hardware

Equipment	**Purpose**	**Notes**
Sony VAIO VPCEB laptop	Data acquisition	
Hewlett Packard 8648C Signal Generator	Transmitter	FRS channel 17
Motorola Talkabout MR350R	Transmitter	FRS channel 17
USRP2	Transmitter	FRS channel 17
CORNET SDR nodes	Sensors	

To transmit and/or receive on the USRP2's required using GNU Radio's software components [57]. For the receiver, or sensor, we modified GNU Radio's `usrp_spectrum_sense.py` to run on a USRP2 and fine-tuned the spectra to be observed. The original Python script was written for a USRP, a related Ettus product. The USRP2 transmitter relied on `benchmark_tx.py`, a standard script featured in GNU Radio.

Experimental Design

The measurements reported in subsequent sections is accomplished by logging the RSS measurements received by the twelve CORNET nodes from a mobile transmitter over a certain period of time. For a particular location, each dedicated SDR node recorded power measurements only if the transmitted signal surpassed a threshold value. The twelve SDR node readings, per location, then would form a fingerprint tuple $[FP_1, FP_2, \ldots, FP_n]$. Note that the recorded power measurements from our `usrp_spectrum_sense.py` are not given in dB or dBm, so they were converted to dB after the measurement campaign.

5.2.2 Statistical Properties of RSSI

Because we chose to measure RSS, there are several factors that immediately affect it. The major factors are the presence/absence of people, antenna orientation, distance from receiver, and time of day. We consider these factors because they certainly affected the RSS measurements as shown in this chapter. Keep in mind that these factors are not all inclusive. A number of experiments are conducted that vary these factors, and the results are analyzed.

Table 5.2 lists the factors into two groups and the options used in subsequent studies. The factors are categorized by what aspect of the fingerprinting process it most affects. We

also note that this study performs measurements on a quasi-static or static transmitter. The assumption we use is that the typical usage of the spectra occurs under stationary conditions. Any movement of the transmitter results in fluctuations of the RSS, or small-scale fading [58]. Large fluctuations of RSS values can also be exploited to determine an mobile station's movement. Nonetheless, mobility of the transmitter is beyond the scope of the current study.

The first factor, orientation of the transmitter, must be considered because it can change the estimated fingerprint location. A different orientation may cause a change in the fingerprint location at the same location. Our second factor emphasizes that different radios have different bandwidths, some more narrower than others. Narrow bandwidths can make it difficult to distinguish fingerprint locations and each fingerprints uniqueness.

Table 5.2: Factors that affect an RSS Fingerprint

Effect on	**Factors**	**Options**
Data Collection	Orientation of transmitter	North, East, South, and West
	Make of transmitter	FRS radio, USRP2, and HP signal generator
Statistics	Time of measurement	times of day
	Period of measurement	Second, minutes
	Interference	Co-channel or adjacent radio channel

Environment changes over time, such as movement and presence of people, around a measurement location is considered. This might indicate the need to collect fingerprints at different times of the day to best reflect the RSS's time dependency. The fifth factor, period of measurement, is needed to justify the minimum duration of a measurement required to obtain a fingerprint at each location. Both of these factors will influence the positioning system's accuracy and the total time required to collect fingerprints. We also consider the sources of interference to signals using the same or different frequency.

Fingerprint Grid

Because of the limited access to the laboratories and offices, the measurements were taken only in the main passageway. The passageway's dimension is 70.12 m (230.05') in length and 2.37 m (7.76') in width. We defined two different grid patterns: one with twenty-two positions and one with forty-five positions. When using the grid pattern of twenty-two positions, the minimum grid spacing, or minimum distance between two locations, was fixed at 3.05 m (10') with a total of 264 RSS distributions (22 points x 12 CORNET nodes). The grid spacing with forty-five locations has a minimum grid spacing of 1.52 m (5') and 540 RSS distributions. Fingerprints were collected at each location for a period of four minutes at a rate of ten samples per second. Each fingerprint position is numbered in ascending order starting with the number '1' with the location closest to the mechanical room. Note that the transmitter's orientation is limited to facing northwest, or facing toward the first

Figure 5.3: Grid pattern of ICTAS' first floor with 22 positions

floor's exit to Stanger Street. Lastly, the sensors, the CORNET SDR nodes, are located in the overhead and placed approximately collinearly.

Figures 5.3 and 5.4 illustrate the grid patterns used for the study. Note that the yellow circles are the approximate locations of the CORNET nodes used. Measurements are taken at the center of each blue rectangle.

Table 5.3 summarizes the measurement scenarios. Scenario 1 through Scenario 3 have similar configurations with the exceptions of the transmitter used, times of measurements, and the minimum grid spacings. The SDR nodes in CORNET cannot sense for extended periods of time, especially hours and days, due to hardware limitations and the high potential of hardware overheating. Hence, Scenario 4 is used to measure the sensitivity of RSS over a

Figure 5.4: Grid pattern of ICTAS' first floor with 45 positions

Table 5.3: Experimental design and measurement factors

Factors	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**
Transmitter orientation	NW	NW	NW	NW, NE, SW, SE
Time of Measurement	Evening	Morning	Evening	Morning
Span of Measurement	1 day	1 day	1 day	1 hour
Period of Measurement per Position	4 minutes	4 minutes	4 minutes	20 minutes
Number of Positions	22	22	45	1
Distance between Positions	1.52 m	1.52 m	3.05 m	N/A
Transmitter	Motorola MR350R	USRP2	USRP2	HP8648C
Number of Sensors	Variable	12	12	12

period of time.

Distribution of RSS

In general, the average RSS in an indoor environment is believed to be lognormally distributed according to the large scaling model [58]. Its mean value is predictable and chiefly follows one of the standardized path loss models (i.e. Nakagami, etc.). However, the studies completed in [24] indicate otherwise. In [24], the RSS distributions were found to be non-Gaussian and asymmetric. Moreover, the RSS distributions could have multiple modes [24].

For Scenarios 1 through 3, we generated 132, 264, and 540 histograms. Observations from the 936 sets of data indicate that certain shapes of distributions occur often at particular average values of RSS. Note that these different shapes of distributions are caused by the upper and lower bound of measurable RSS at each location. Note that the RSS will never reach a value that is as high as the maximum transmitted power because received signal attenuates over distance. The histograms of Scenario 1 are in Appendix A and Scenario 2 and 3 are in Appendix C. Note that the histograms in Scenario 2 look similar to the even numbered locations in Scenario 3 despite taking the measurements at a different time. Instead, we will use Scenario 3 for much of the upcoming analysis.

We also compare the skewness of each data set with its standard error of skewness. The standard error of skewness is estimated by $\sqrt{\frac{6}{N}}$, where N is the number of data points [59]. The data set is considered to be significantly skewed when the the absolute value of skewness is larger than two standard errors of skewness. In Scenario 1, we found 76 histograms were significantly left-skewed and 47 were significantly right-skewed. In Scenario 3, we observed that 391 histograms were significantly left-skewed, while only 138 were significantly

right-skewed. From these results, the histograms with significant left-skewed data generally correspond to a strong average RSS or a line-of-sight exists between the transmitter and receiver.

Figures 5.5 and 5.6 show the skewness of all histograms in Scenario 1 and Scenario 3. Notice that both have rather large negative skewness. However, data with low RSS values generally had skewness around the zero value, while data with higher RSS values have large negative skewness. These skewed distributions often occur when the RSS level is low due to large distances between the transmitter and receiver and/or the lack of a line-of-sight path.

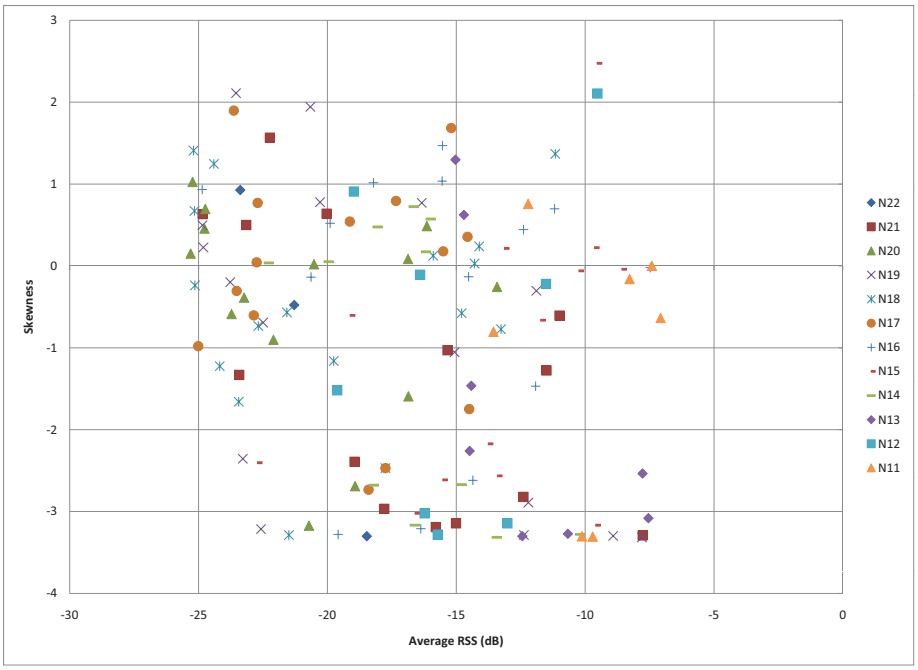

Figure 5.5: Average RSS vs. skewness in Scenario 1

Another aspect of the RSS measurements we analyze is the standard deviation. The plots shown in Figures 5.7 and 5.8 confirm that the farther the sensor is located or the lower the RSS is will result in small standard deviations. When the average RSS is large, the standard deviation is also large. Note that the property suggests that RSS from two locations may be difficult to distinguish if both locations are close to the same sensor. Furthermore, two locations may be better identified with NLOS paths.

The last study quantifies the independence of multiple signals of each RSS fingerprint by calculating the correlation coefficient between any two sequences of received signals per location. Recall that the correlation coefficient is a real number between 0 and 1, calculated

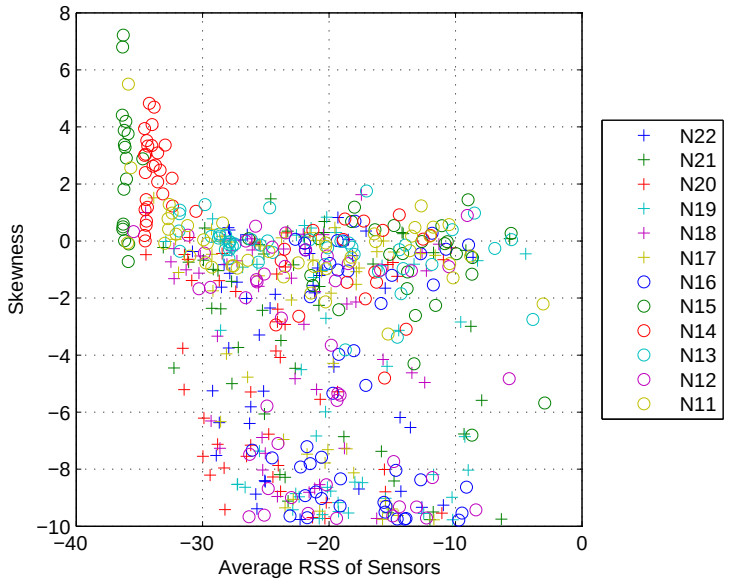

Figure 5.6: Average RSS vs. skewness in Scenario 3

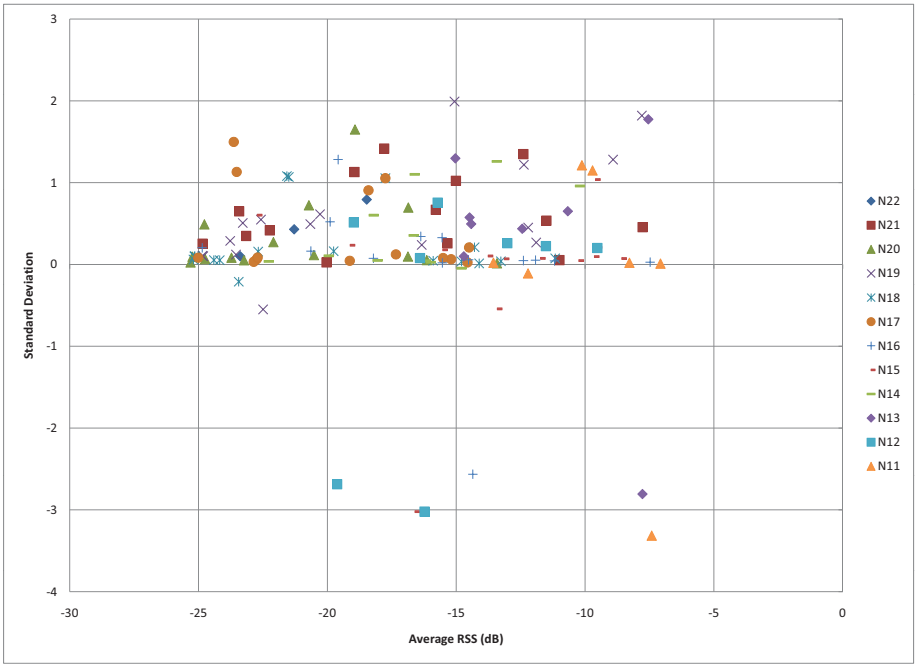

Figure 5.7: Average RSS vs. standard deviation for Scenario 1

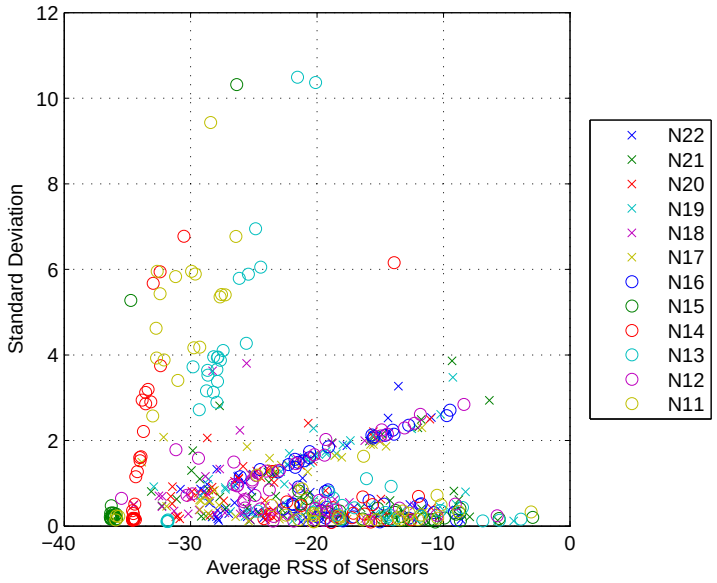

Figure 5.8: Average RSS vs. standard deviation for Scenario 3

Table 5.4: Correlation coefficient interpretations

Correlation Coefficient	Description
0.0 - 0.1	trivial, very small, insubstational, tiny
0.1 - 0.3	small, low, minor
0.3 - 0.5	moderate, medium
0.5 - 0.7	large, high, major
0.7 - 0.9	very large, very high
0.9 - 1.0	nearly, practically

from $R = \frac{\sigma_{i,j}^2}{\sqrt{\sigma_i^2 \sigma_j^2}}$. Here $\sigma_{i,j}^2$ is the covariance between random variable i and j and σ_i^2 and σ_j^2 are the variance of random variable i and j, respectively. If two signals are not correlated or have no effect on each other, the correlation coefficient will be 0. If two signals are dependent on one another or have a strong relationship, the correlation coefficient approaches the value of 1. From [59], we use the following guidelines to classify the correlation coefficient of two random variables in Table 5.4.

In Appendix D, the tables show correlation coefficients for a small subset of sensors. In Table D.1 and Table D.2, the correlation coefficients significantly were not consistent. This results implies that the transmitter power was not constant over the time of measurement and the signal may be too narrow in bandwidth. Table D.3 has more consistent correlation coefficient values, which means that increasing the transmitter's bandwidth significantly helped. The results in Table D.4 suggest low correlation coefficients for the sensors coincide with the

sensors receiving a severely degradated signal due to distance or a non line-of-sight path. Note this type of RSS behavior is expected in the indoor environment.

5.2.3 Causes of Error

The randomness of RSS indoors is a major cause of error in an RSS-based indoor positioning system. No RSS variability or randomness would result in no location detection errors or excellent accuracy and precision performance. Here, we investigate the causes of error in identifying indoor locations. First, we will discuss the error based on the results of our statistical analysis in the previous section. Second, we analyze the effects of the separation of RSS patterns with different physical distance separations and different numbers of sensors used.

Randomness of RSS Patterns

The randomness of the RSS measurements is clearly described by is probability density function (PDF) or distribution. The stronger the mean RSS is, the more its left tail distribution will be large. In comparison, weaker mean RSS values tend to have symmetric distributions.

Since RSS occasionally represents the physical distance between the sensor and transmitter, location detection errors are more likely if this distance is larger. The closer the transmitter is to the sensor sometimes can make localizing the position more difficult. One possible reason for this is the non-linear mapping between the actual RF energy and the reported RSS values of each sensor. Intuitively, weaker RSS values will be harder for the sensors to differentiate the signal of interest and its receiver noise. Ideally, if all the sensors could measure small RF energy well, we would expect the RSS distributions to be left-skewed across all mean RSS values.

Separation of Fingerprints

The performance of fingerprint-based localization systems largely depends on the granularity of the separation of fingerprints. A distinct fingerprint corresponding to one location can be identified correctly if it is difficult to incorrectly classify it as another fingerprint by a pattern classifier. In this section, several plots are examined to visualize how position granularity affects creating fingerprints.

Figures 5.9, 5.10, and 5.11 show the variability of the RSS for each position. Figure 5.12 shows the variability of the RSS for only one fingerprint position over an extended period of time. Notice that for positions in the middle of the hallway the RSS values are very close. But these figures do not give a clear picture as to the underlying RSS patterns. To more

closely investigate how the patterns of RSS at different locations may appear, we consider RSS samples at four locations for Scenario 3.

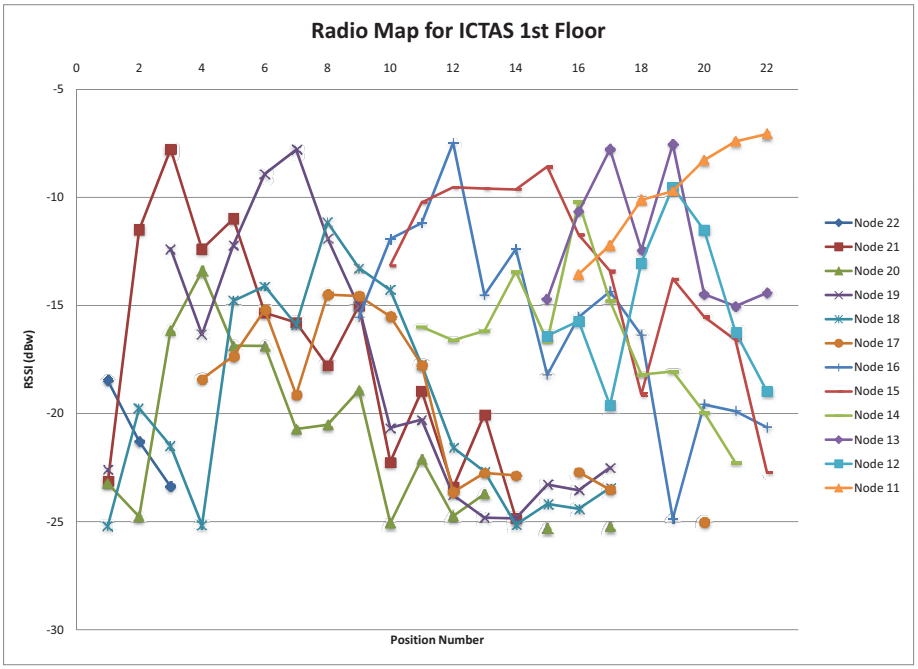

Figure 5.9: Variability of RSS for Scenario 1

Figure 5.13 is a two dimensional plot of samples from two sensors, N22 and N21. The group of patterns at each location can be called the fingerprint of that particular location. From the plot, patterns of each location is grouped together as a cluster. Notice that the spreading pattern is not symmetric, which is due to the RSS's left-skewness. If it were symmetric, the clusters would be perfect circles. Clearly the RSS patterns in Figure 5.13 can be separated by some discriminant function or clustering technique. However, there are smaller groups of patterns located further away from the main clusters. Notice also that the RSS at position 27 and 28 have overlapping patterns. This suggests that having two sensors is not enough to discriminate between fingerprints.

Figures 5.14 and 5.15 show a similar plot as Figure 5.13 but with patterns from multiple pairs of sensors. Using multiple pairs of sensors can significantly help in separating fingerprints. However, having too many sensors could possibly not improve location estimation, especially when the signal has large variations.

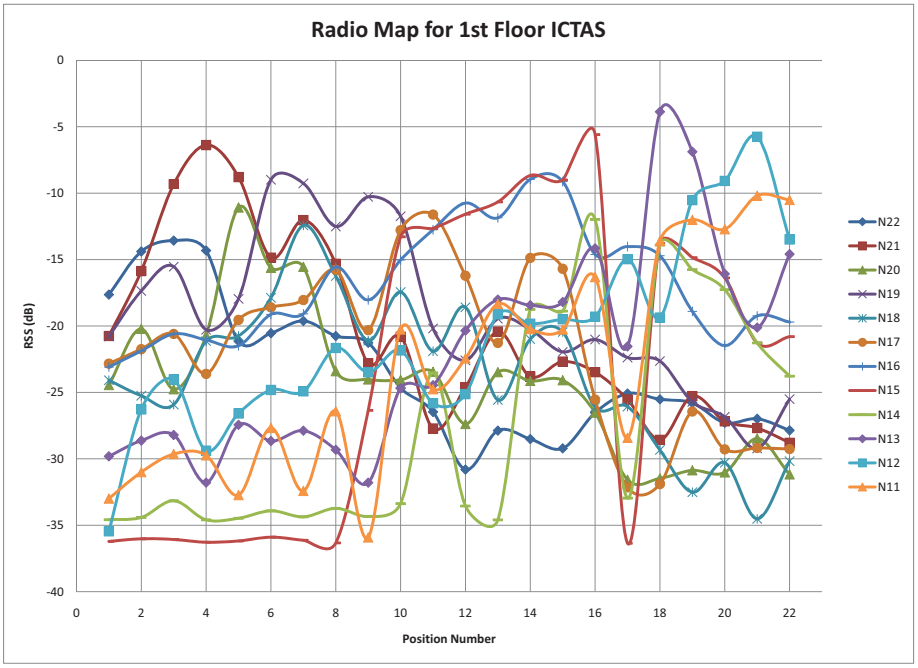

Figure 5.10: Variability of RSS for Scenario 2

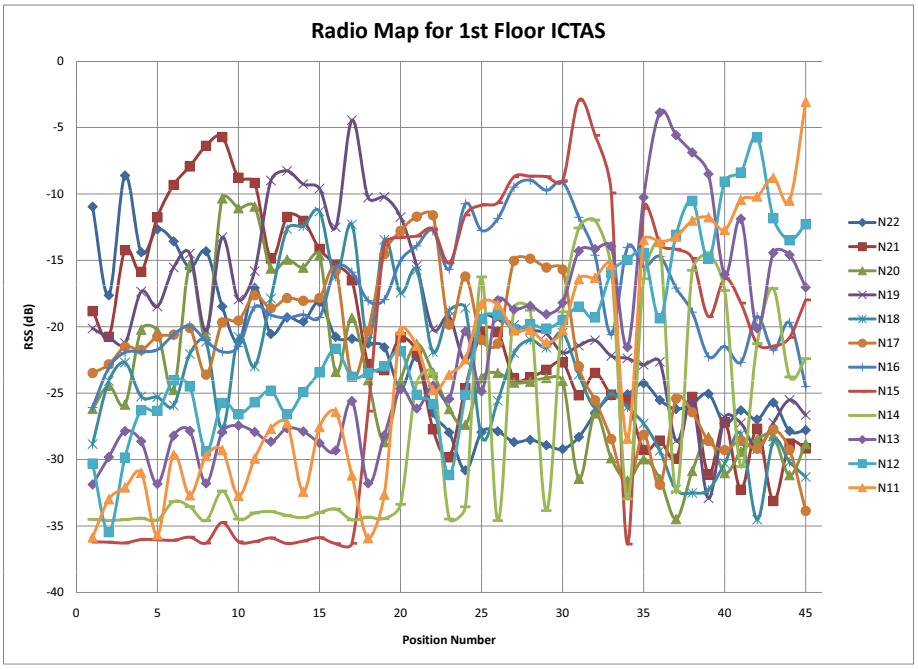

Figure 5.11: Variability of RSS for Scenario 3

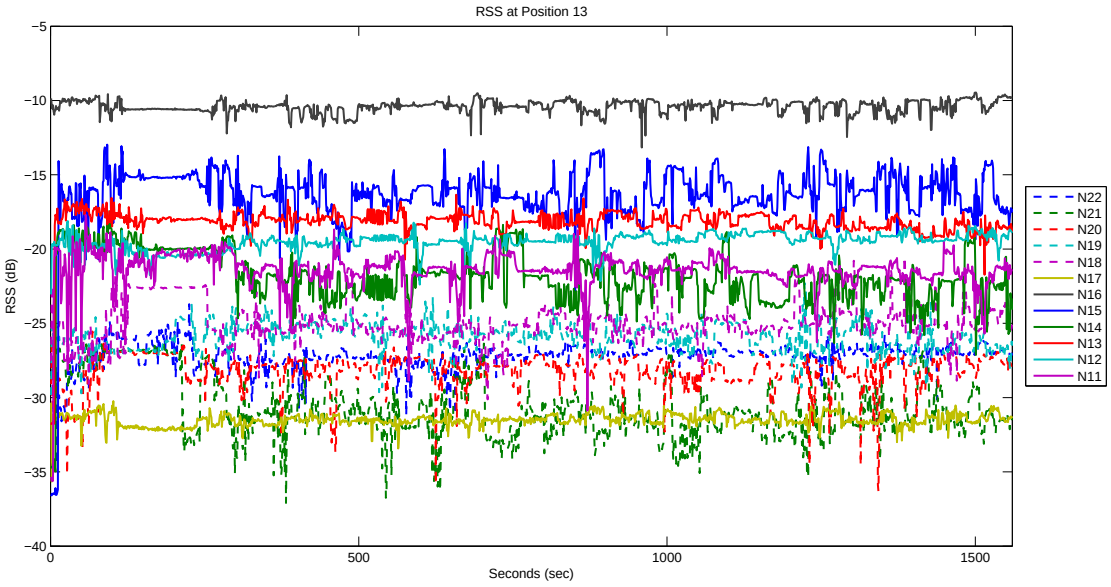

Figure 5.12: Variability of RSS for Scenario 4

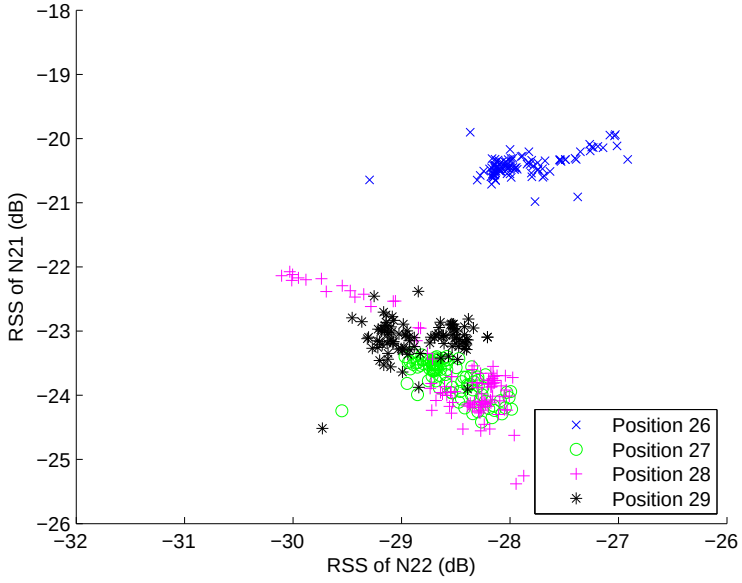

Figure 5.13: Spreading patterns of different locations

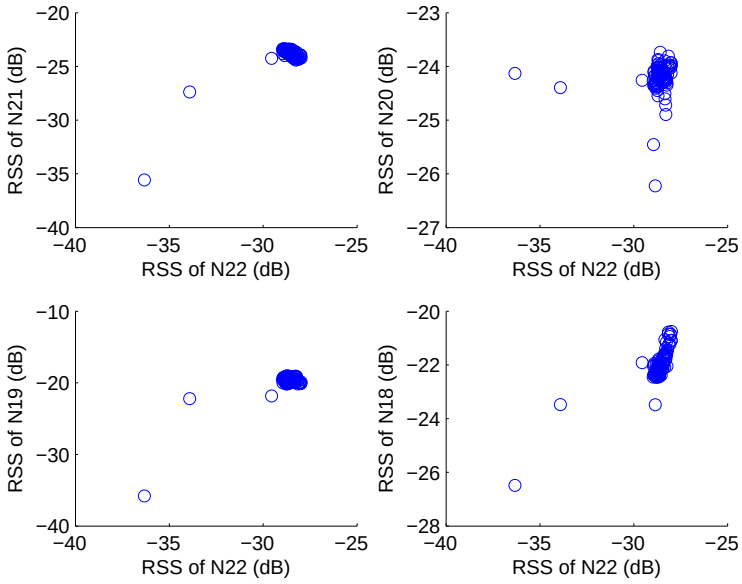

Figure 5.14: Scatterplot of position 27 from Scenario 3

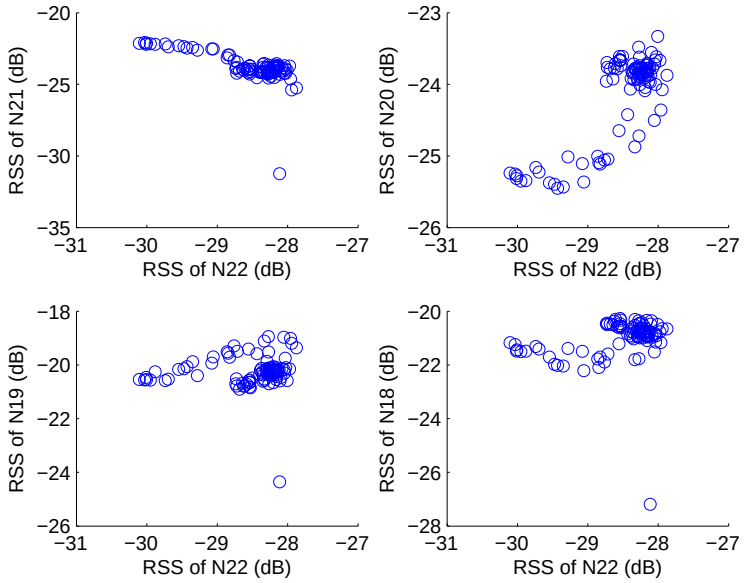

Figure 5.15: Scatterplot of position 28 from Scenario 3

5.2.4 Implications on Modeling

To implement a fingerprint-based indoor positioning system, we must consider some mathematical assumptions for the fingerprints. We will list the assumptions that are derived from the analysis above.

- We assume that the random RSS is a stationary process. This eliminates the time dependency properties of the RSS.
- The mean RSS of each RSS is assumed to be constant for each location and corresponding fingerprint.
- All signals received by the sensors are assumed independent. Although, the correlation between each signal can be represented by a covariance σ_{ij} in a covariance matrix, we could use a multi-variate normal distribution. Instead, we will use the empirical PDFs from the RSS values.
- RSS values comprising a fingerprint are time invariant.

5.3 Testing of Distributed Localization Algorithms

Prior to implementing the fingerprint-based positioning system, the algorithms discussed in Section 4.4 must be tested for robustness and accuracy. Simulations of DEDA, DBDA, and DNNA were performed in MATLAB on a laptop computer with an 2.53 GHz Intel Core i5 processor and 4 GB of memory. Each algorithm is implemented in three phases: (1) single thread on a single processor, (2) multiple threads on a single processor, and (3) multiple threads on multiple processors. The RSS values used as input to the algorithms are from the measurement campaigns from Scenario 2 and Scenario 3. The next subsections will highlight the localization performance of each proposed algorithm in the multithread, single processor phase.

Note that each algorithm is subjected to four transmission scenarios: (1) walking from southeast to northwest, (2) walking from northwest to southeast, (3) short transmission, and (4) sporadic transmission. These transmission scenarios have sequential transmissions and do not assume the transmitter "backtracks" on a position. Note that transmission scenario three has the user begin transmitting at a position other than position 1 and stops transmitting well before the last fingerprint position. In transmission scenario four, the user intermittently transmits.

Table 5.5 lists the AORs assigned for each thread. Observe that the number of positions per AOR is not homogeneous, which is intentional.

Table 5.5: AORs for multiple threads

	Thread 1	**Thread 2**	**Thread 3**	**Thread 4**
Scenario 2	Position 1-3	Position 4-10	Position 11-17	Position 18-22
Scenario 3	Position 1-11	Position 12-22	Position 23-33	Position 34-45

5.3.1 DEDA Simulations

Recall that DEDA partitions the fingerprint database into p partitions and assigns each node a partition for processing, where p_i for $i \in \{1, 2, \ldots, p\}$ is the assigned partition. Each p_i contains a subset f_j of the f fingerprints.

Shown in Figures 5.16 to 5.19 are example path trajectories taken by a POI using only twenty-two fingerprint positions. Figures 5.20 to 5.23 have the path trajectories using forty-five fingerprint positions. The top subplot in each figure presents the results of the single thread, single processor phase. The bottom subplot reveals the outcomes of the multiple thread, single processor phase.

A common trend in these plots is how the same positions are incorrectly estimated. The positions most often missed when using a twenty-two position radio map are position 8,

position 13, and position 18. A closer look at the radio map with twenty positions affirms that position 8 and position 9; position 13 and position 14; and position 18 and position 19 have very similar fingerprints. When using the radio map with forty-five positions, positions 14, 19, 24, 29, 34, and 41 are missed. We attribute the incorrect positions again to having non-unique fingerprints with adjacent positions.

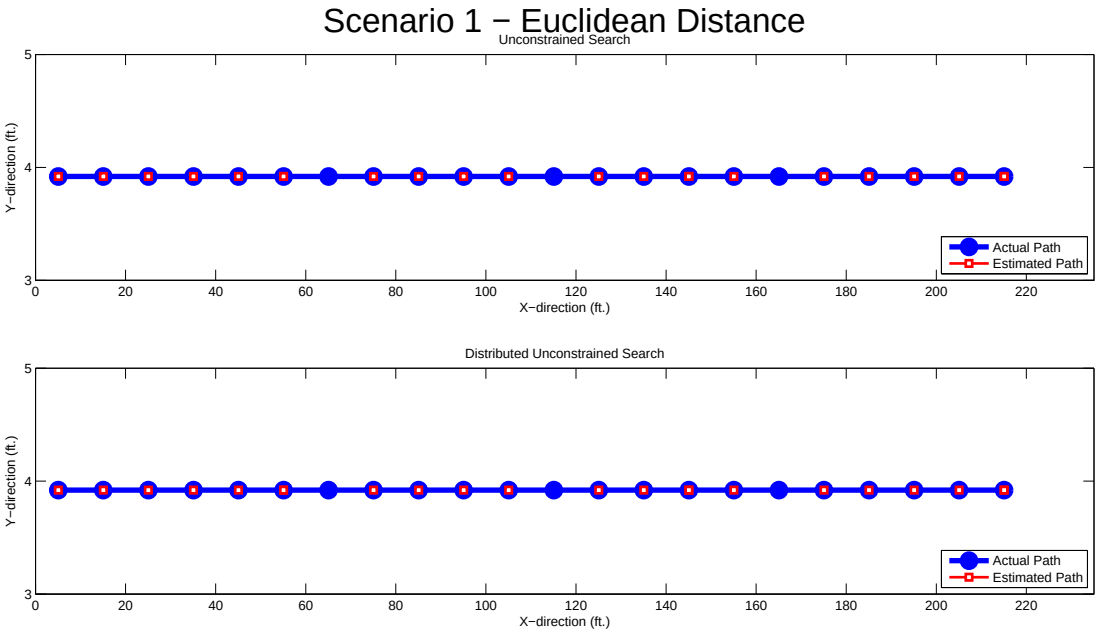

Figure 5.16: DEDA - Transmission scenario 1 with 22 positions

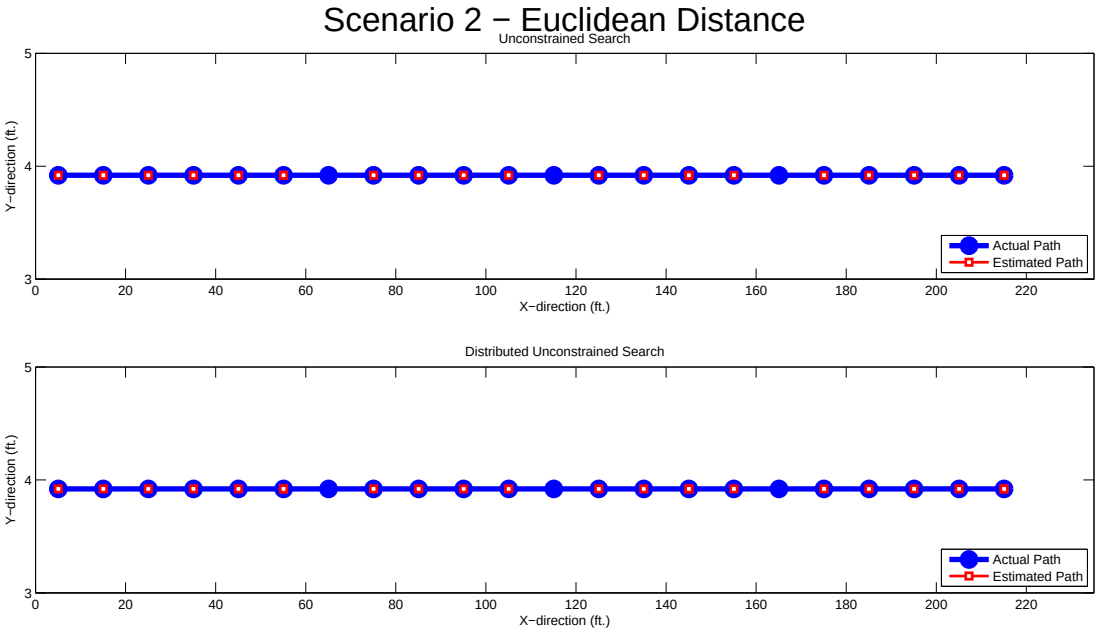

Figure 5.17: DEDA - Transmission scenario 2 with 22 positions

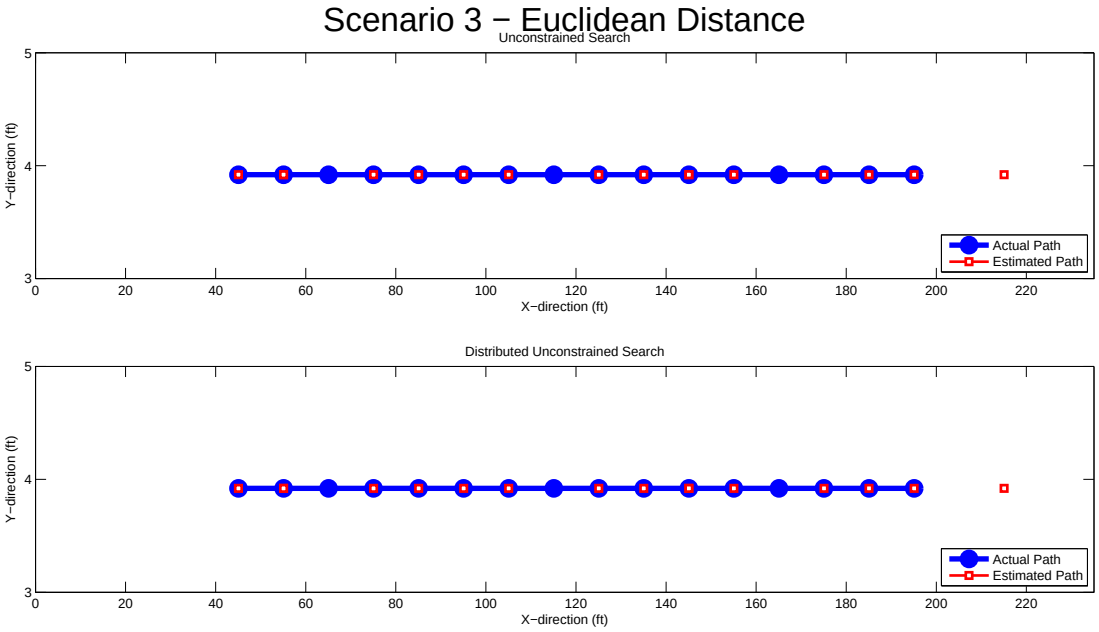

Figure 5.18: DEDA - Transmission scenario 3 with 22 positions

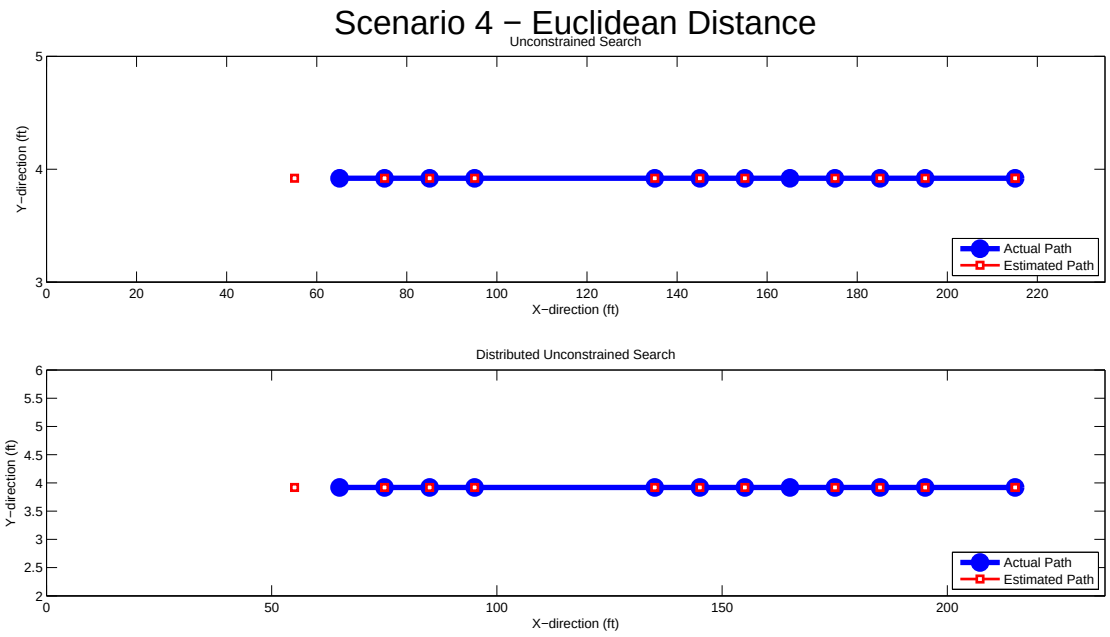

Figure 5.19: DEDA - Transmission scenario 4 with 22 positions

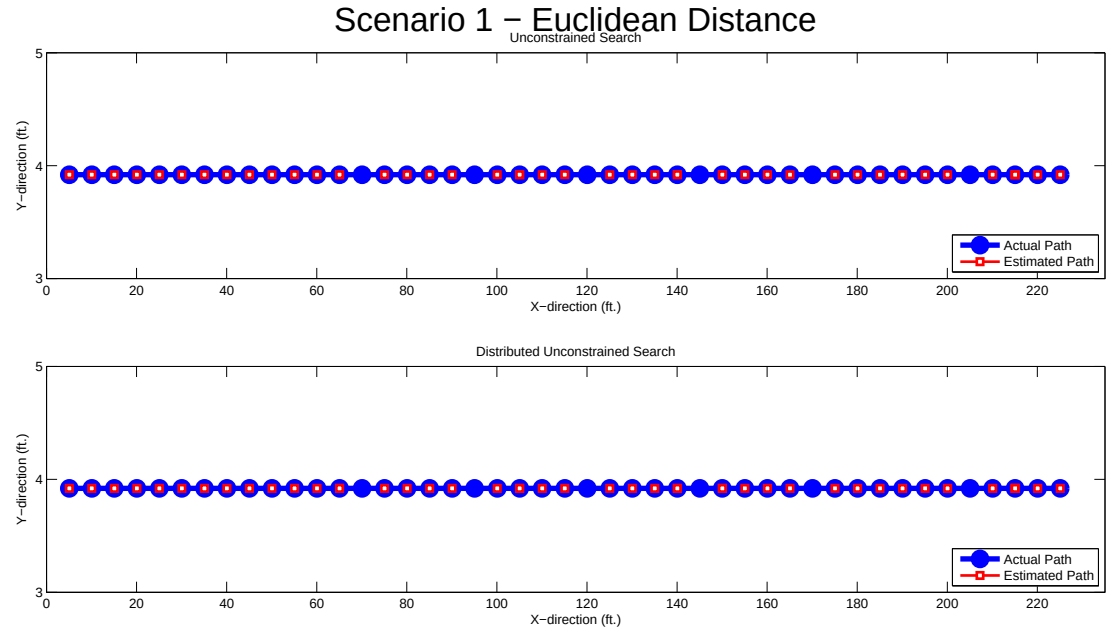

Figure 5.20: DEDA - Tranmission scenario 1 with 45 positions

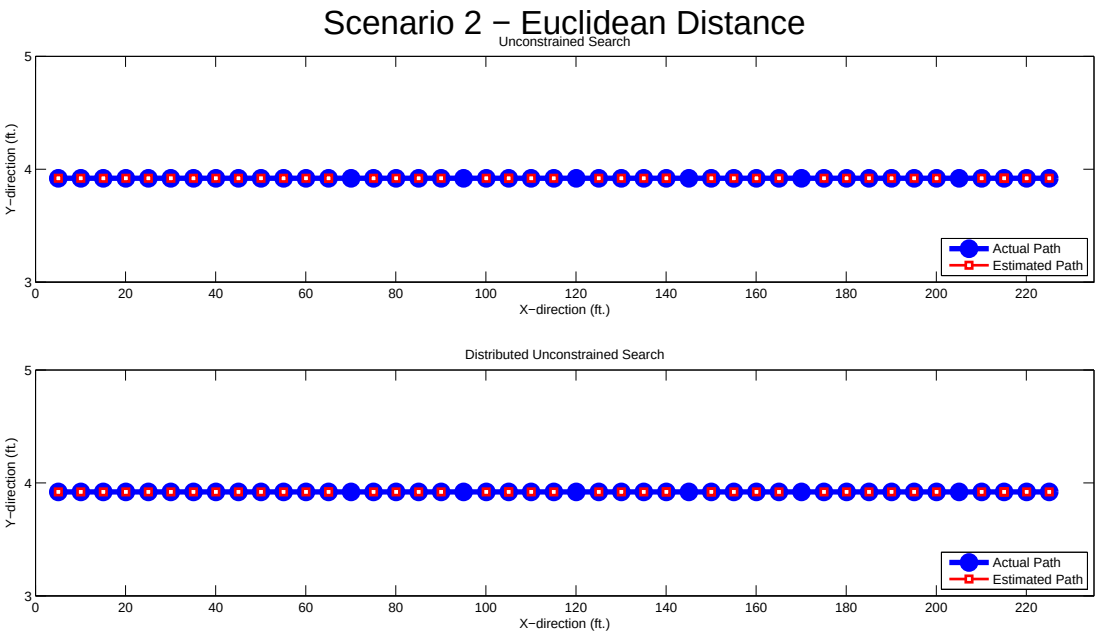

Figure 5.21: DEDA - Tranmission scenario 2 with 45 positions

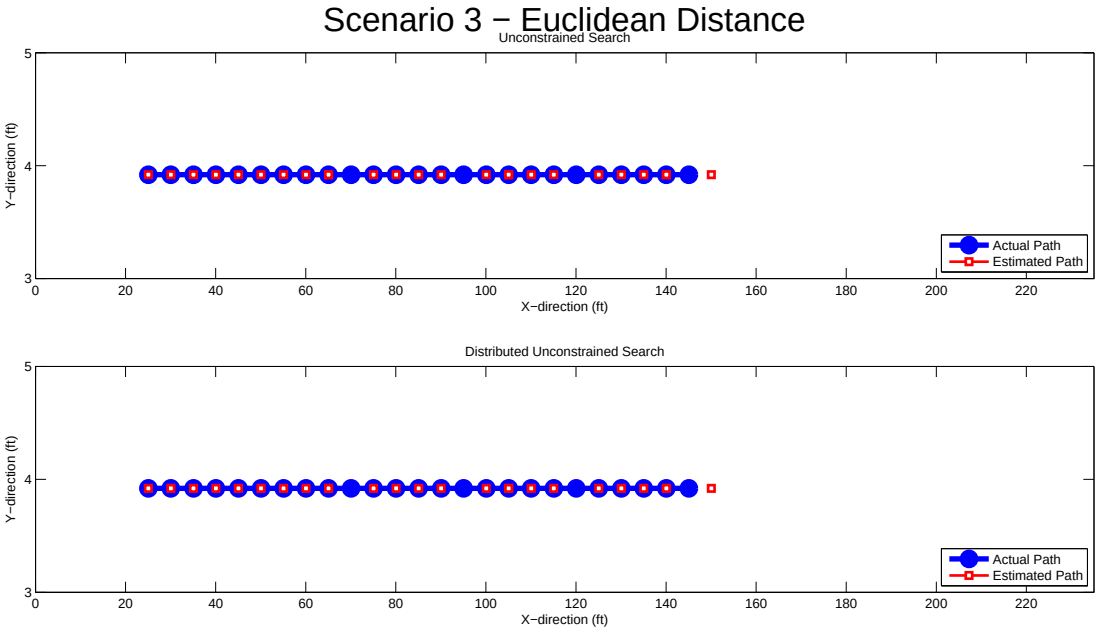

Figure 5.22: DEDA - Tranmission scenario 3 with 45 positions

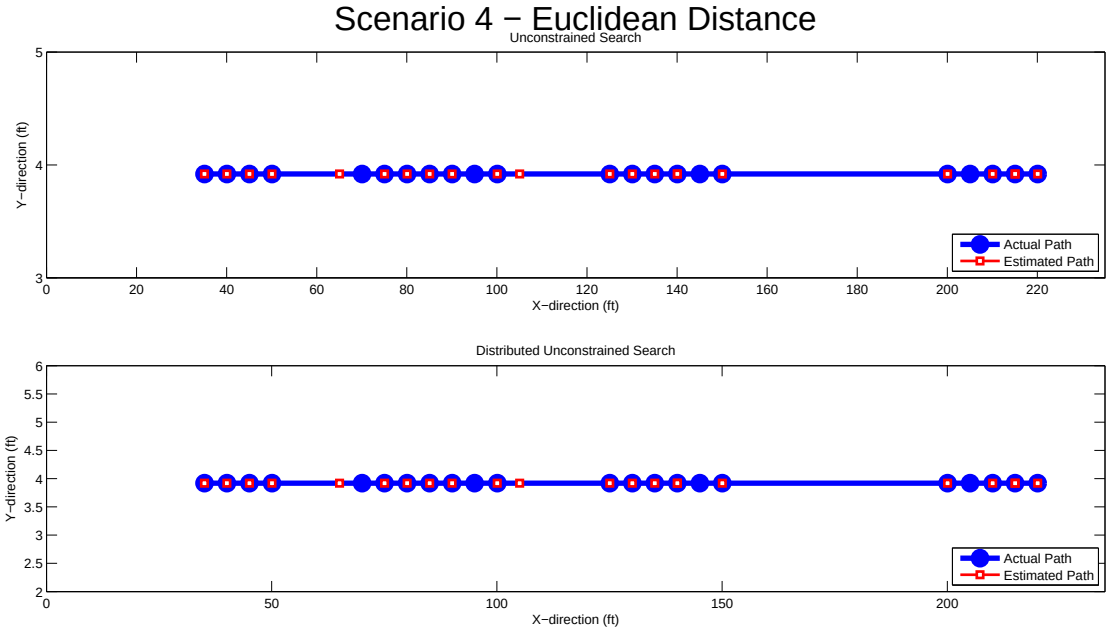

Figure 5.23: DEDA - Tranmission scenario 4 with 45 positions

5.3.2 DBDA Simulations

The DBDA is computationally expensive because it continually solves for Equation (2.16). Furthermore, it incorporates the user's motion in predicting where he/she will move next, which should improve performance. In these simulations, we use the most general case in which the user is equally likely to move to any location within a region of a given distance from the previously predicted location. The empirical PDF's are used in solving Equation (2.16).

Like in the previous section, certain positions are incorrectly identified. For the radio map with twenty-two positions, positions 13 and 21 are missed. Positions 26, 29, and 42 are incorrectly estimated when using the radio map with forty-five positions.

For this particular trial run, DBDA performed better than DEDA. However, when DBDA incorrectly estimated a position and the actual location was outside the search radius, DBDA predictions become inaccurate. Furthermore, a constrained search space yielded recovering from poor decisions unlikely.

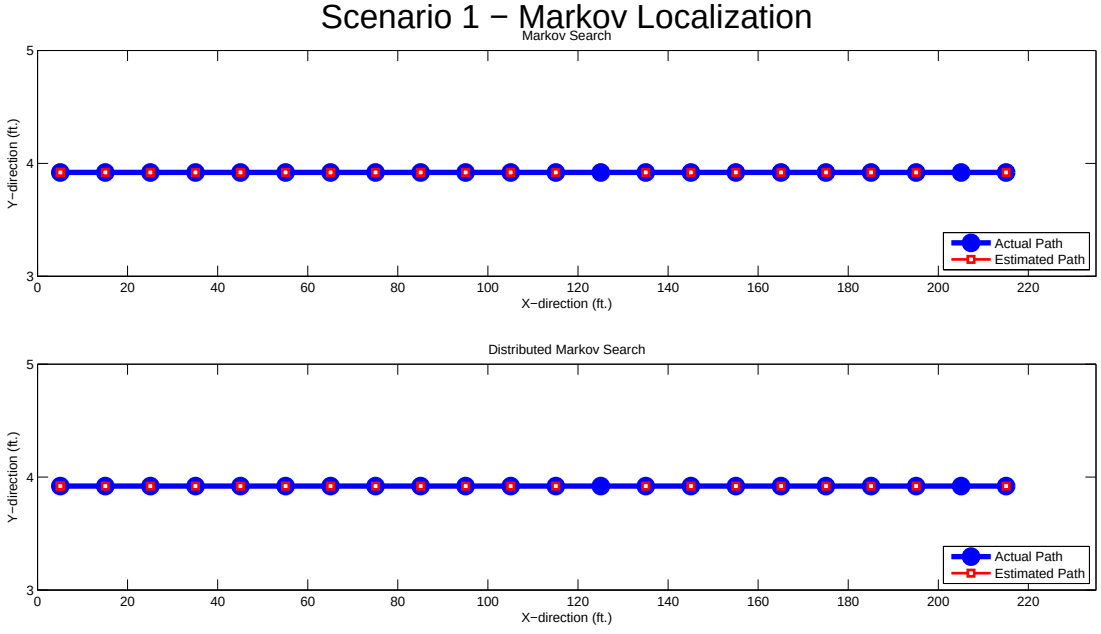

Figure 5.24: DBDA - Transmission scenario 1 with 22 positions

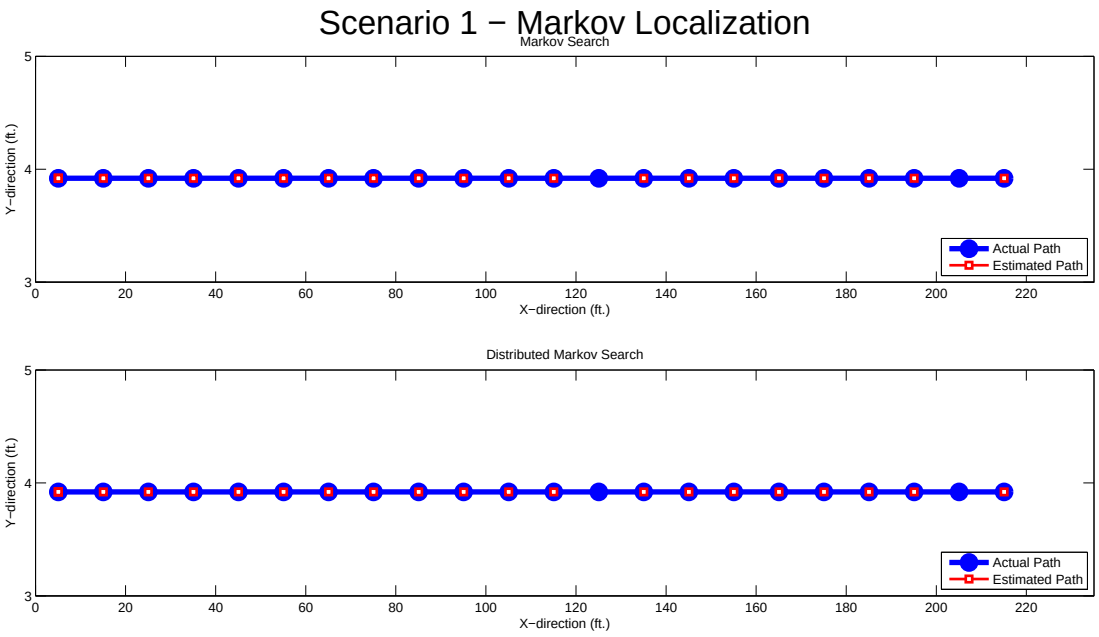

Figure 5.25: DBDA - Transmission scenario 2 with 22 positions

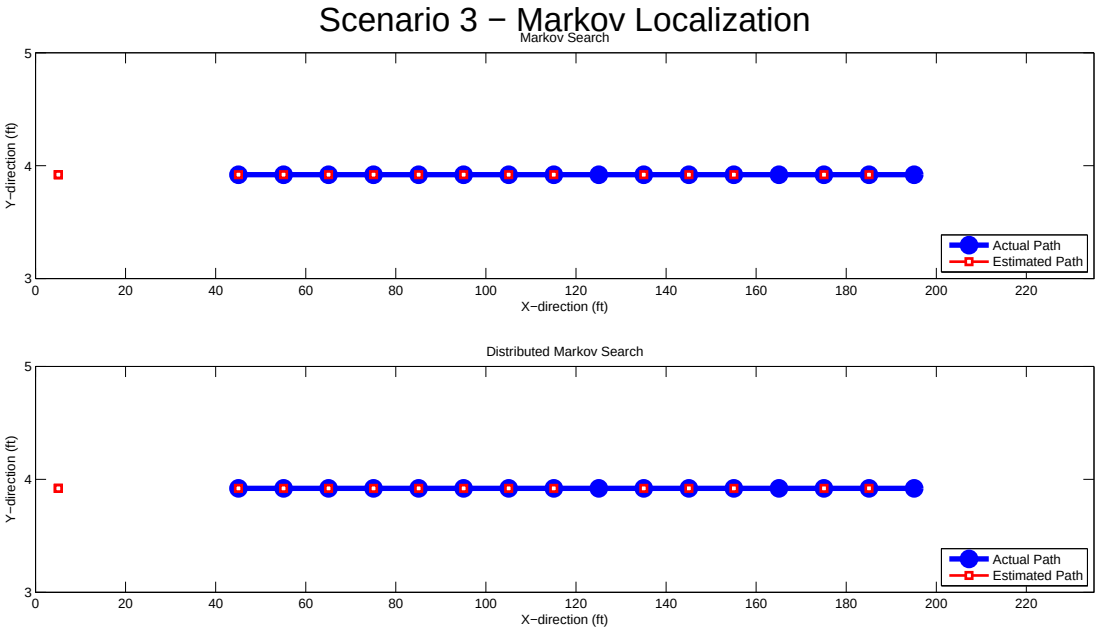

Figure 5.26: DBDA - Transmission scenario 3 with 22 positions

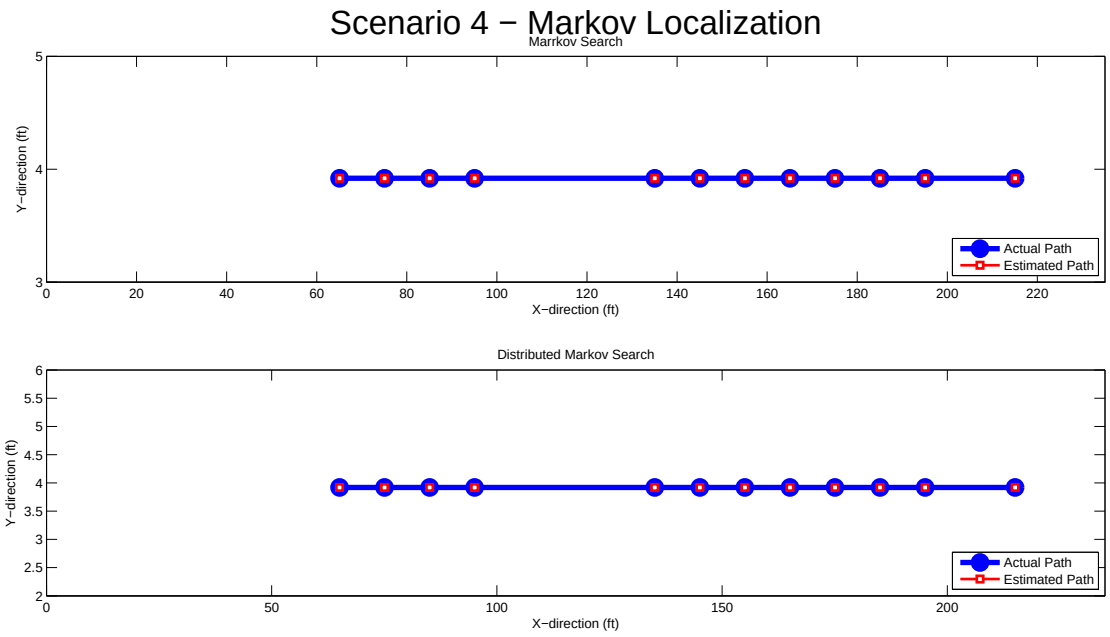

Figure 5.27: DBDA - Transmission scenario 4 with 22 positions

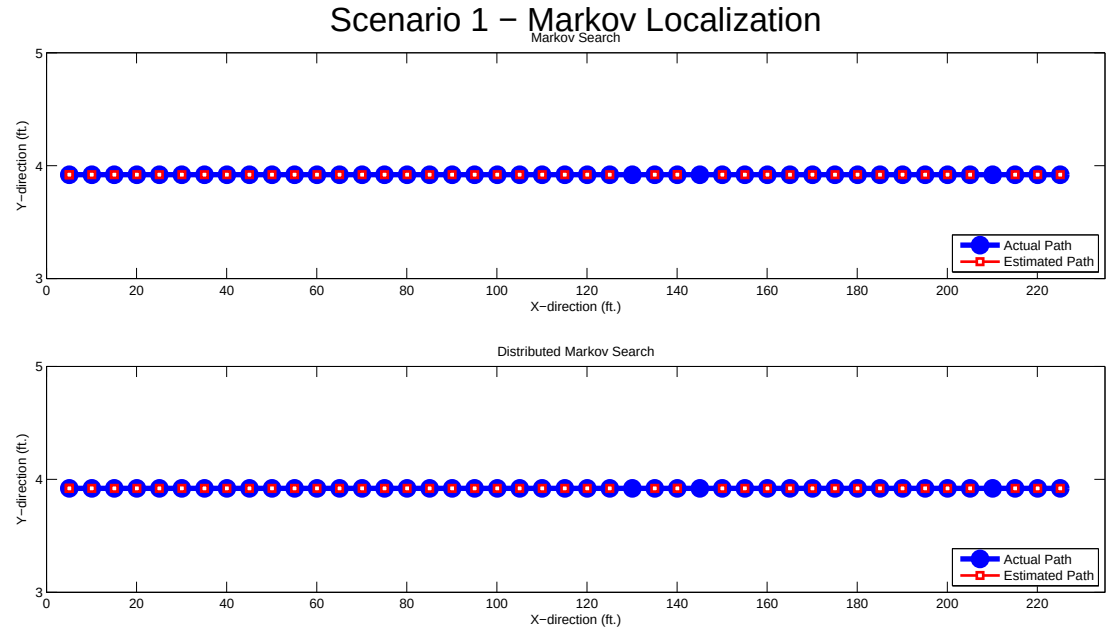

Figure 5.28: DBDA - Transmission scenario 1 with 45 positions

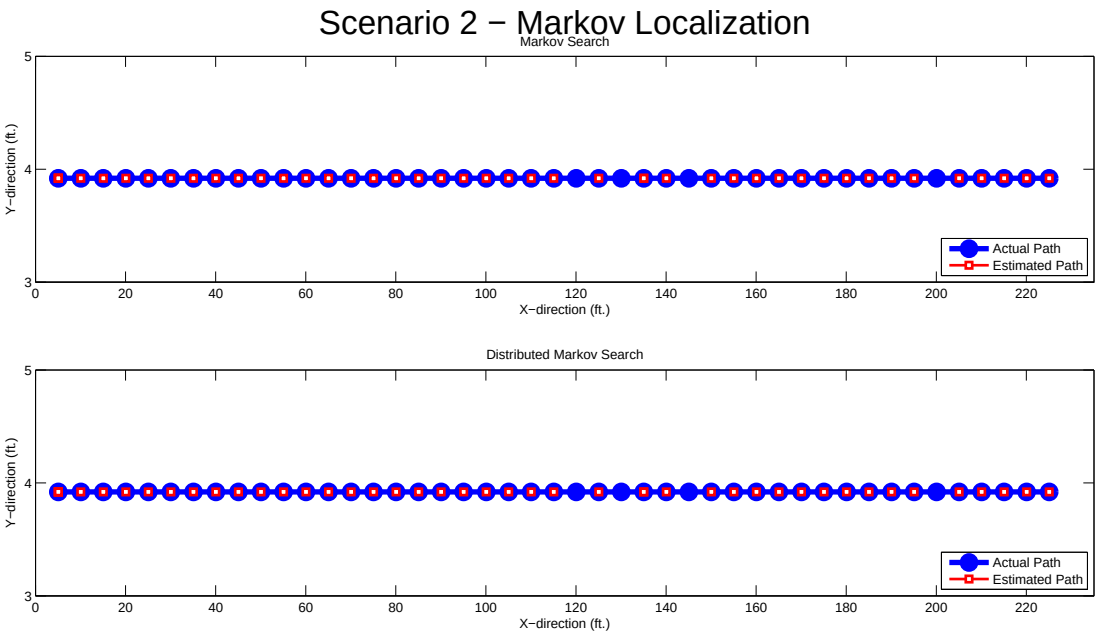

Figure 5.29: DBDA - Transmission scenario 2 with 45 positions

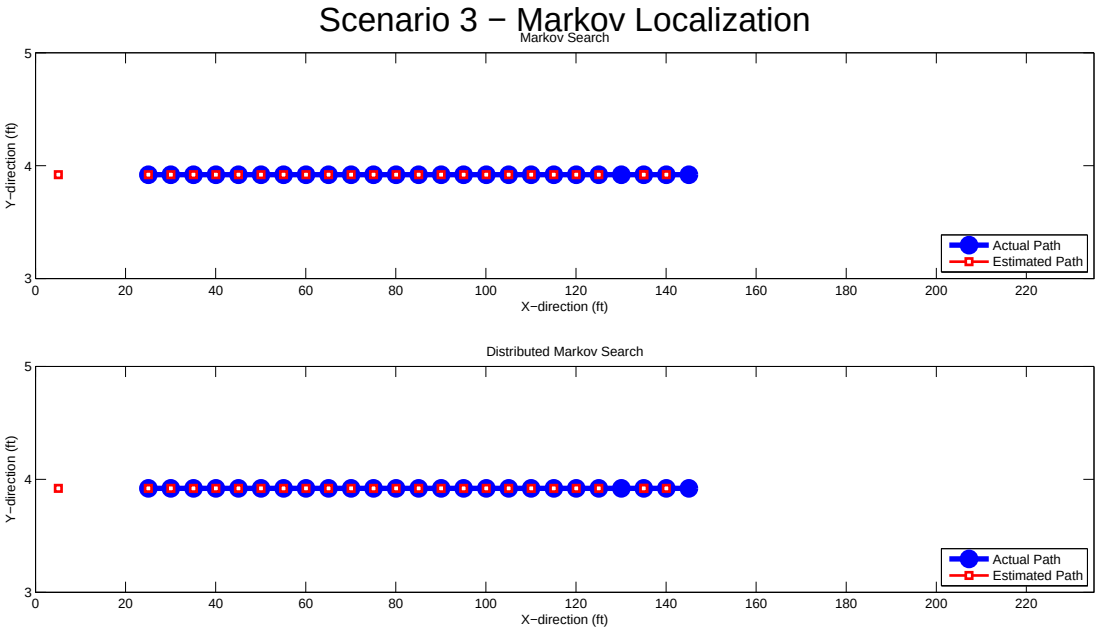

Figure 5.30: DBDA - Transmission scenario 3 with 45 positions

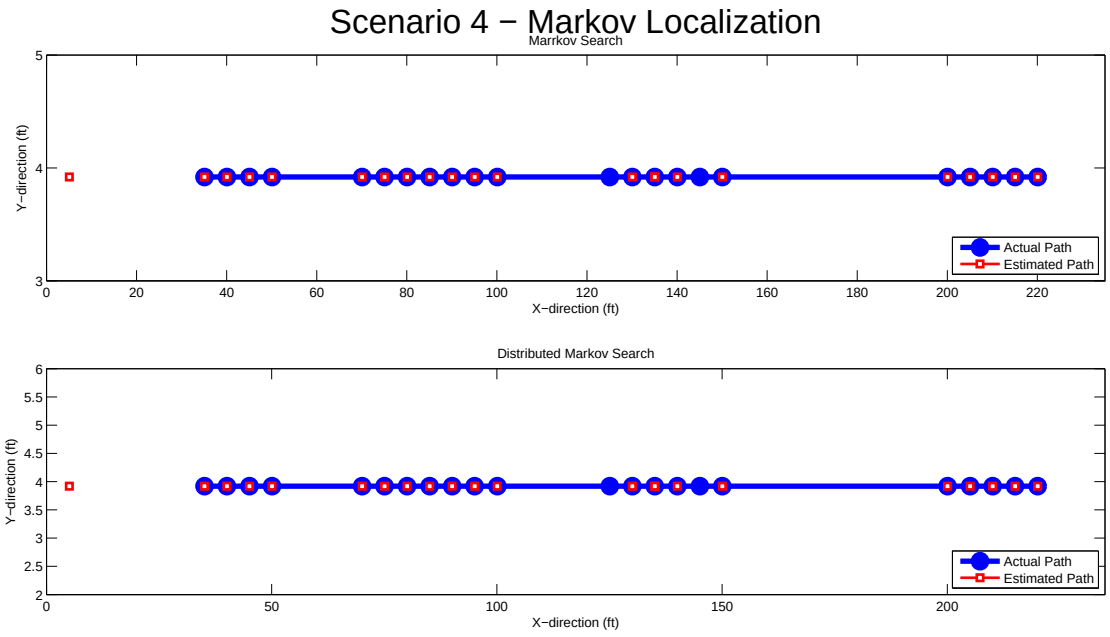

Figure 5.31: DBDA - Transmission scenario 4 with 45 positions

5.3.3 DNNA Simulations

The MATLAB simulation environment contains a suite of neural network related functions required to setup and test a distributed neural network. We trained and tested two types of neural networks: a generalized regression neural network (GRNN) and a multilayer perceptron neural network (MLP). The first network tested consisted of four GRNN's corresponding to the AOR's listed in Table 5.5. The second network had four MLP neural networks and similar AOR's.

Note that the successful classification of fingerprints stems from the critical offline training of each network. Figures 5.32 to 5.35 plot the minimum mean square errors (MMSE) of the offline and online phases of each MLP neural networks. The MLP neural network in Figure 5.32 was trained with only 30 data sets. Its average MMSE is 0.8124 m (2.665'). Figure 5.33 used 50 data sets, and the average MMSE decreased to 0.2987 m (0.9799'). When trained with 75 and 100 data sets, Figure 5.34 and Figure 5.35 indicated an average MMSE of 0.2908 m (0.9541') and 0.3488 m (1.1454') respectively. Thus, increasing the number of training information, in general, did decrease the average MMSE. To be consistent, we chose to use 100 training sets for both the MLP neural network and GRNN.

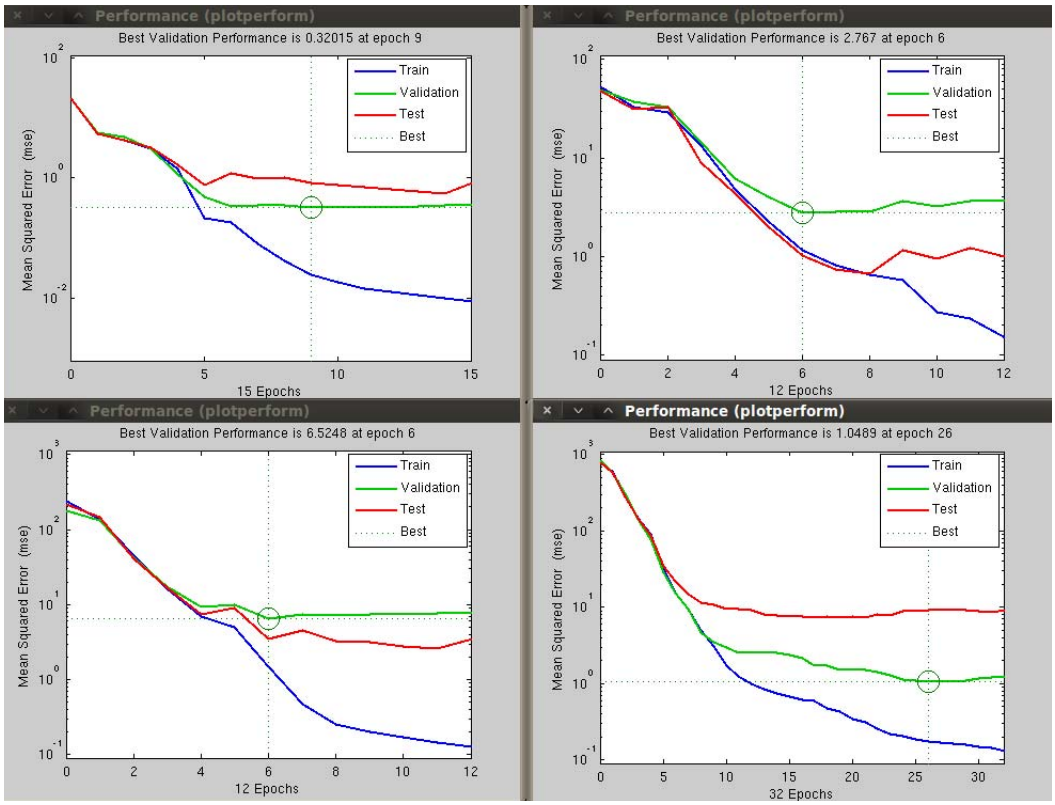

Figure 5.32: Minimum mean square error of a MLP neural network with 30 sets of training data

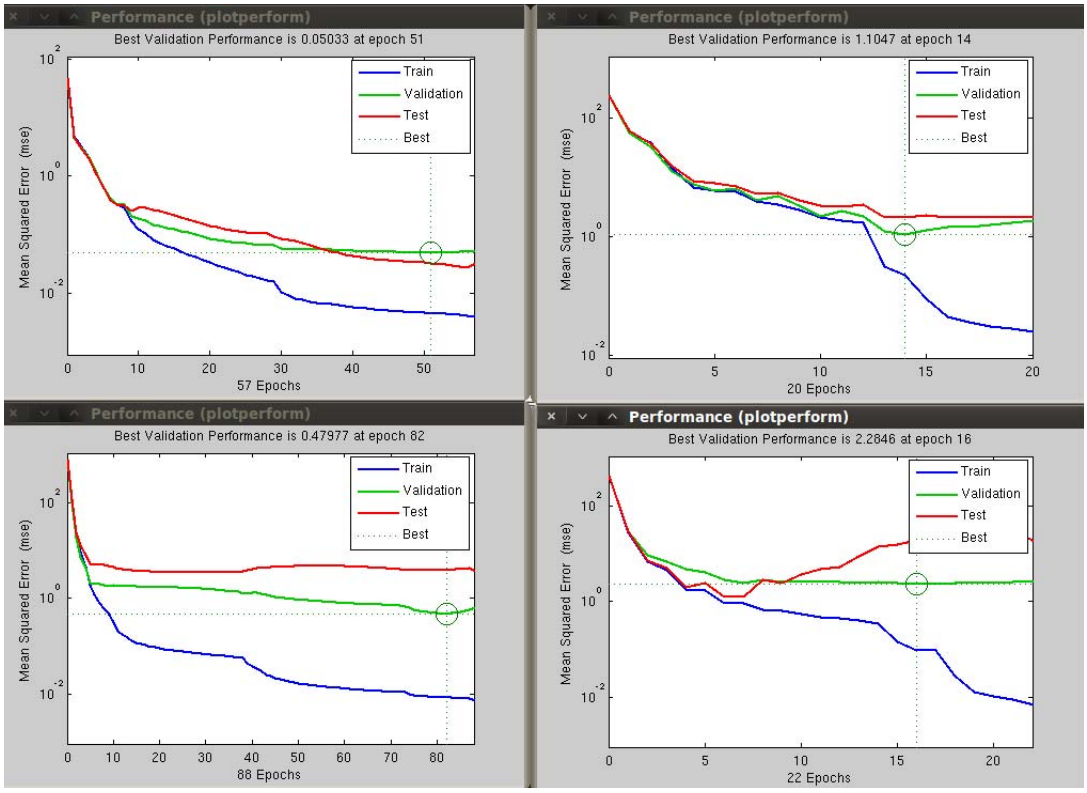

Figure 5.33: Minimum mean square error of a MLP neural network with 50 sets of training data

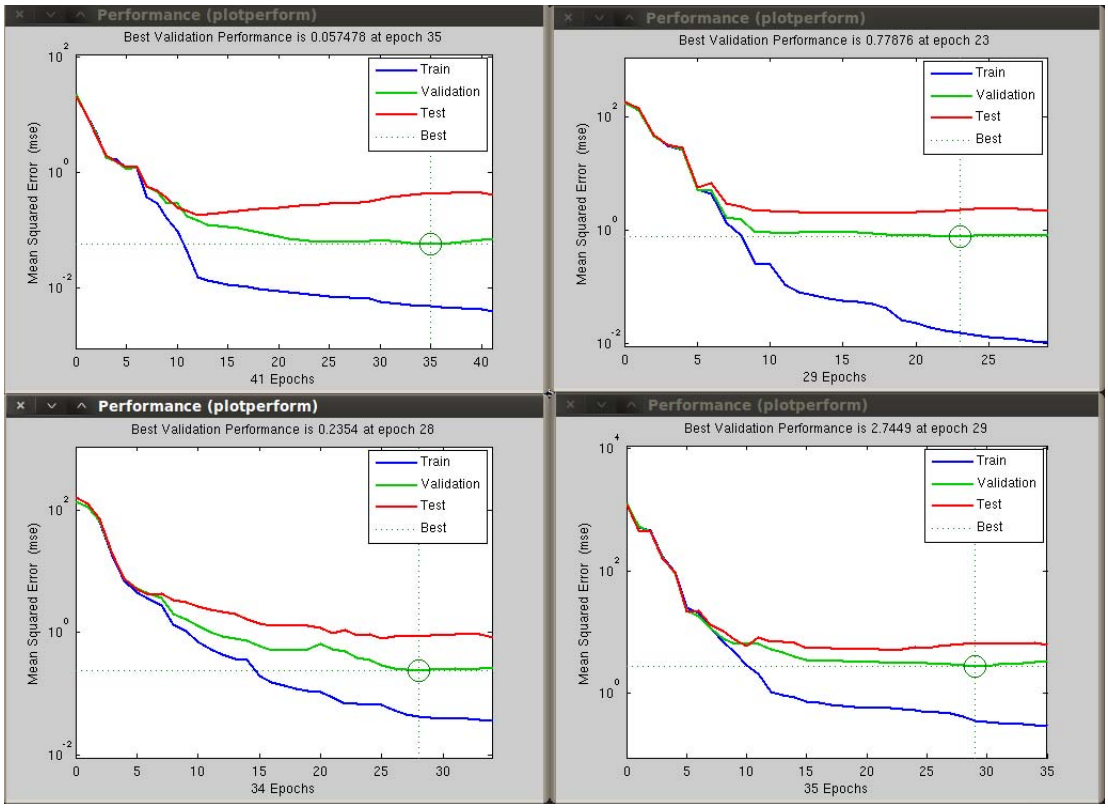

Figure 5.34: Minimum mean square error of a MLP neural network with 75 sets of training data

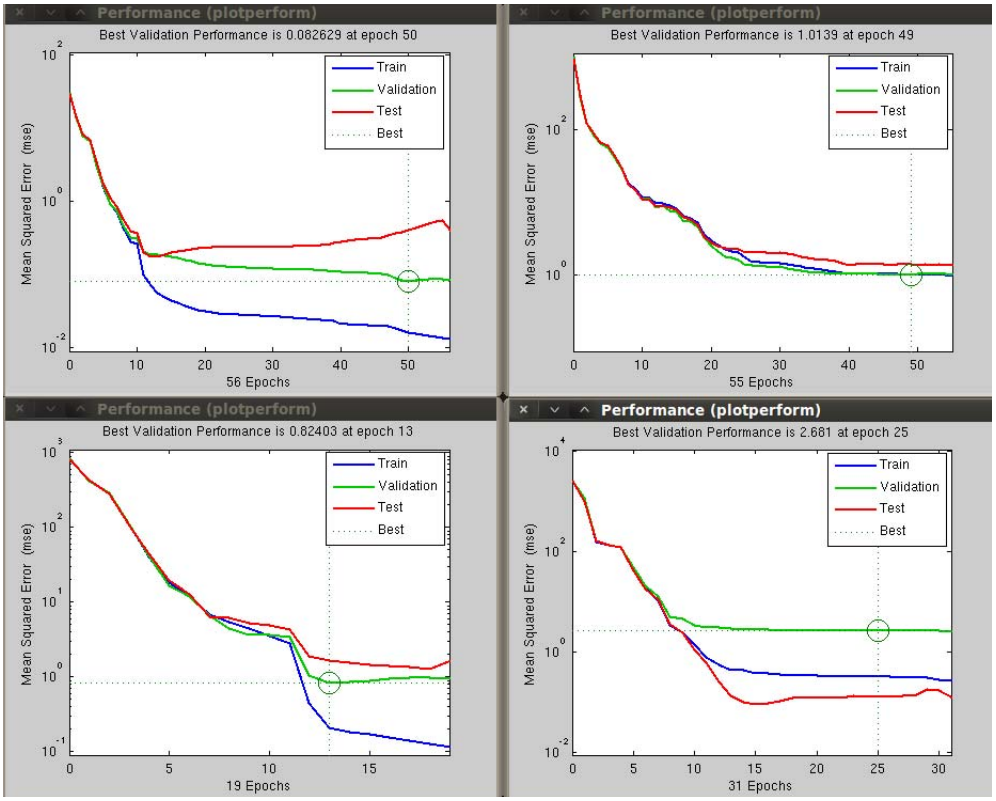

Figure 5.35: Minimum mean square error of a MLP neural network with 100 sets of training data

5.3.4 Performance Analysis

This section only measures the performance of transmission scenario one, but the results can be easily extended for the other transmission scenarios. Our results display the average localization errors for 1000 runs of 100 experimental data sets.

Show in Figure 5.36 are the results of one sample run. The subplots are arranged in the following order: DEDA (upper left), DBMA (upper right), MLP (lower right), and GRNN (lower left). Recall that the blue circles are the actual path taken by the user and the red squares are the algorithm's position estimate. Of the four distributed localization algorithms, the GRNN's pattern matching algorithm performed superbly and achieved a minimum localization error of 0%. The other distributed algorithms performance order, from best to worst, is DBMA, DEDA, and the MLP neural network. One possible reason for the MLP neural network's poor performance is due to the algorithm's solutions to the non-linear mapping of fingerprints to positions. Unlike the GRNN, the estimates by the MLP neural network are not necessarily discrete values. Therefore, the estimates must be rounded up (ceil) or rounded down (floor). Then obtaining a discrete value becomes inherently difficult.

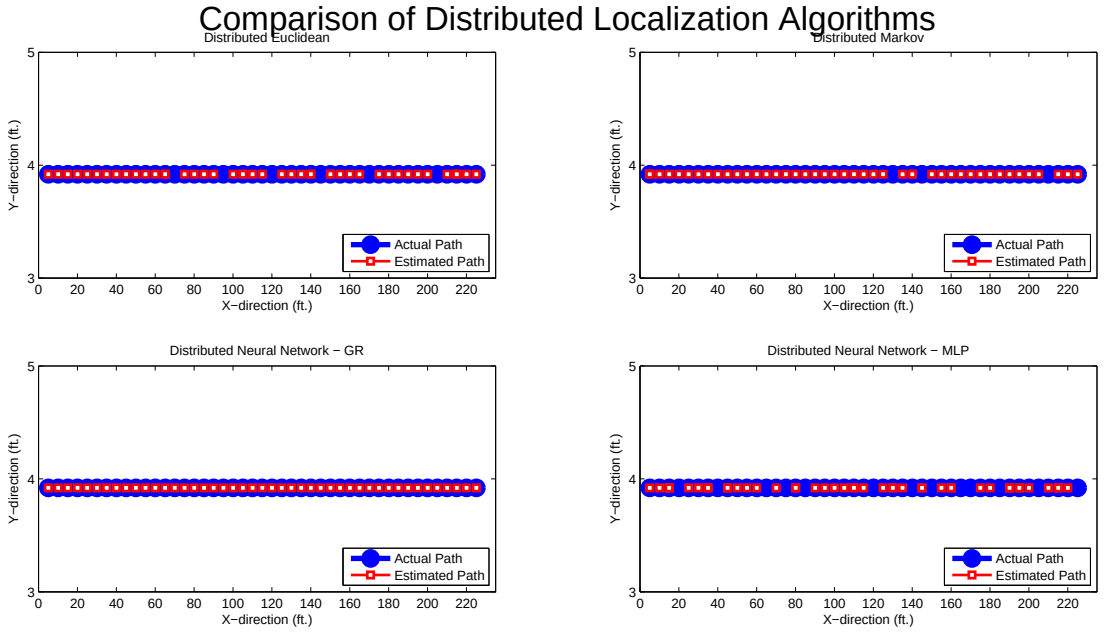

Figure 5.36: Comparison of transmission scenario 1 for all distributed localization algorithms

To better compare each algorithm's performance, Figure 5.37 has the histogram of location errors for each algorithm for one test run. The subplots are in the following order: DEDA (upper left), DBMA (upper right), MLP (lower right), and GRNN (lower left). Because the user's trajectory is linear and does not vary along the y-direction, the error radius is in fact the location errors in the x-direction. In the MLP histogram of Figure 5.37, the user is more

likely to be estimated within 1.524 m (5') of his or her position than at the actual position. Such a high probability arose from the user's general motion model. Using a more defined motion model may result in the DBMA performing better.

Figure 5.38 presents the cumulative distribution functions (CDF's) of the histograms in Figure 5.37. The blue, red, green, and black lines correspond to the algorithms, respectively, in this order: DEDA, DBMA, GRNN, and MLP. For DEDA, a location accuracy of 3.048 m (10') is found for 74% of the empirical data. This accuracy is comparable to GRNN's location accuracy of 3.048 m (10') of 71% and MLP's location accuracy of 3.048 m (10') of 70%. The worst performing algorithm is DBMA with 30% location accuracy of 3.048 m (10'). To put things into perspective, we look at how our algorithms meet the FCC's E-911's location accuracy standards. To achieve E-911's requirement of 100 m accuracy for 67% of wireless phone calls, our distributed algorithms can achieve this with 87% for MLP, 90% accuracy for both DEDA and GRNN, and 94% for DBMA. Obtaining more precise position estimations would be best achieved, based on our data, using either DEDA or GRNN.

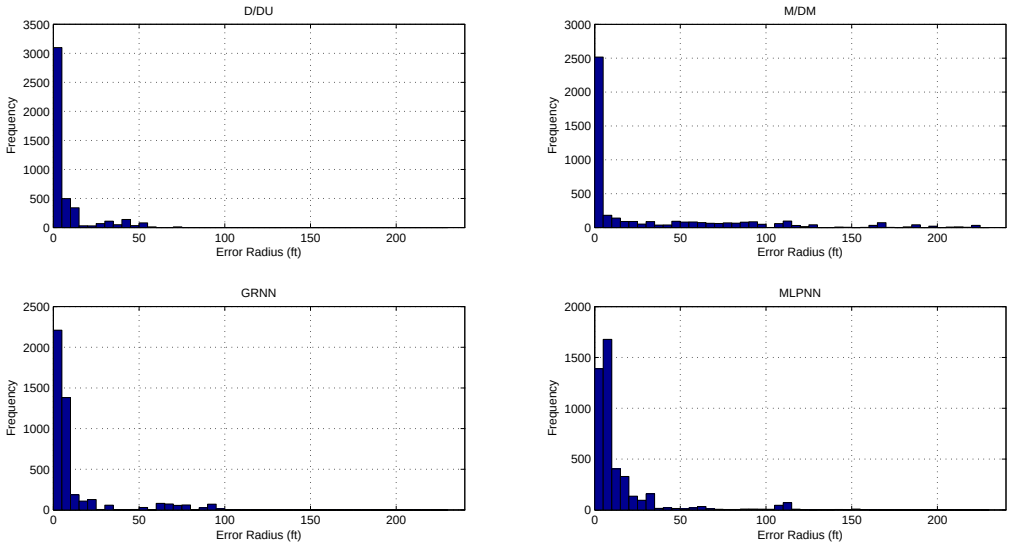

Figure 5.37: Histograms of location error

The results of the 1000 simulation runs is summarized in Table 5.6. The average mean location error for the four algorithms discussed is 5.9424 m (19.4961'). Furthermore, note that the average max error is 40.005 m (131.25'). DEDA has the lowest mean and max position error. The worst mean and max position error occurred when DBMA is used. When comparing the average achievable accuracy of these distributed algorithms to GPS, each perform equally or better than GPS's daily accuracy of 10 m. Hence, the algorithms prove to be great alternatives to GPS.

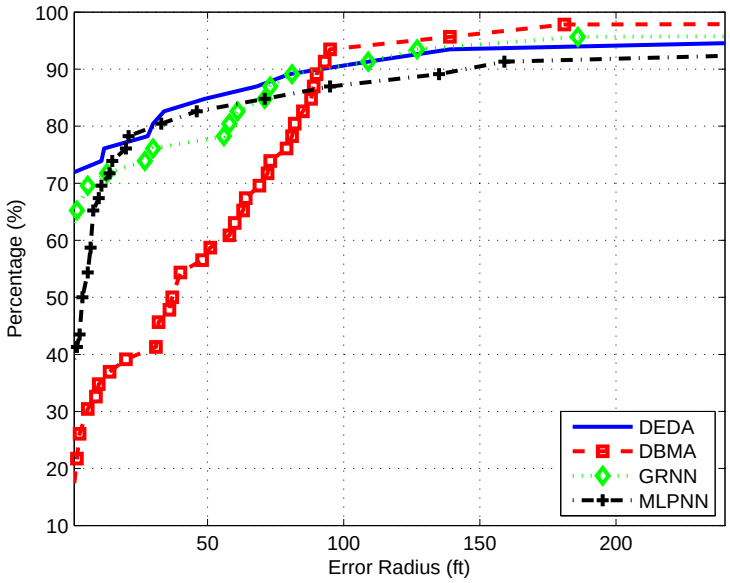

Figure 5.38: Performance comparison of distributed localization algorithms

Table 5.6: Error statistics of distributed localization algorithms

Algorithm	Minimum Error	Mean Error	Max Error
DEDA	0 m (0')	3.2942 m (10.8078')	16.764 m (55')
DBMA	0 m (0')	10.7676 m (35.3267')	67.056 m (220')
GRNN	0 m (0')	4.5568 m (14.9500')	28.956 m (95')
MLP	0 m (0')	5.1511 m (16.900')	47.244 m (155')
Average	0 m (0')	5.9424 m (19.4961')	40.005 m (131.25')

5.4 Conclusion

This chapter focused on two major aspects of achieving a WDC-based fingerprint position system: the RSS of the fingerprint radio maps and the algorithms used to correlate the fingerprints to positions. The measurement campaigns in ICTAS emphasized the time variability and difficulty in applying statistical models to indoor RSS. Since we chose not to model the RSS with a particular probability distribution, our algorithms could be tested in a straightforward manner. The four distributed localization algorithms tested were DEDA, DBMA, GRNN, and MLP. Based on our simulation results, the two best performing algorithms are DEDA and GRNN.

Chapter 6

WDC-based Fingerprint Positioning System

6.1 Introduction

While there exists several research and commercial prototype WLAN-based indoor positioning systems, no system has used software defined radios as sensors and computational nodes to cover large geographic areas. Furthermore, there has not been a WDC-based network developed for position location. Position systems, as mentioned in Section 4.3, either use existing communications infrastructures or require specialized and/or additional technologies. Examples of indoor positioning systems using existing WLAN infrastructures include RADAR [23] and Placelab [60]. Active Badge [61], Cricket [62], SpotON [12], and LANMARC [63] use specialized equipment such as active badges, ultrasound and RFID tags. What makes our positioning system unique is that the position information is not centrally computed at one server. Position information is *wirelessly distributed* to other computing devices in real-time.

The study in this chapter discusses the system design and implementation considerations in deploying our prototype fingerprinting positioning system in ICTAS. First, Section 6.2 gives a brief high level overview of the system design and deployment choices derived from literature. Second, Section 6.3 describes the low level design issues including the selection of distributed positioning algorithms. Thirdly, Section 6.4 explains the functional work and data flow in the WDCN. Fourthly, Section 6.5 demonstrates the use of the WDC-based positioning system and its performance. Lastly, Section 6.6 summarizes the chapter's study.

6.2 High Level System Design

Before deploying our fingerprint-based positioning system, there are several top level issues we considered. The top system design issues included the application, complexity, performance, and security. Each issue dictates how and where components will be located in the positioning system. We give a brief explanation of these system design issues and our choices. Lastly, we discuss the high level description of our fingerprint testbed.

6.2.1 System Design Issues

One of the first design decisions we examined was the position application. Generally, fingerprint-based positioning systems provide locating and tracking services to users inside a building as in [61]. Note that the best accuracy of these systems is along the order of a few meters of the actual location. Centimeter-level accuracy cannot be achieved using indoor RSS values.

Because we are only interested in the performing position location in the passageway of ICTAS, the application's service area is well defined. This greatly limits the service area needing calibration, but will not necessarily reduce the time required for calibration. A related design issue is the granularity of the grid spacing and location resolution. Choosing a fine grid spacing can potentially not improve the system's performance, so we chose to use only 45 positions with a grid spacing of 1.524 m (5'). Furthermore, the size of the grid pattern directly affects the measurement period, complexity, and size of fingerprint radio map. Note that we suspect that increasing the granularity of the grid spacing may actually decrease the system's performance in ICTAS, especially where RSS values are very close as in found in Section 5.2. Other non-trivial system design issues we examined are the number of sensor nodes to use, number of users, the user's orientation, and the number of samples used to estimate a position. These design issues will definitely affect the scalability of the positioning system. Our prototype testbed only locates a single user, traversing the passageway linearly, with a single sample from each sensor. Then we were able to use all twelve CORNET nodes on the first floor for sensing, since they are located collinearly in the overhead.

6.2.2 System Description

Since fingerprinting systems are application-driven, we chose to simulate a patient tracking service for nurses and caretakers at a hospital. Specifically, we consider the scenario where the local hospital is requesting a non-intrusive system capable of monitoring and localizing high-risk patients trying to escape their hospital room. The location information must be accurate for hospital caretakers and security personnel while maintaining a high performance level to reliably monitor the patient's activity.

To provide this location service, our fingerprint-based positioning system encompasses two subsystems: a distributed sensing component and the WDCN. The distributed sensing aspect only senses the user while the WDCN actually computes the user's estimated position with a distributed localization algorithm. The total number of CORNET nodes used to support the application is 17 nodes. The distributed sensing subsystem uses twelve nodes, and the WDCN has only five nodes. We choose to use separate nodes for the position computation to facilitate troubleshooting and minimize destructive communication interference between sensors and the WDCN. One master node and four slave nodes comprise the WDCN subsystem. Once the user's position is estimated, a web-based interface displays the position information.

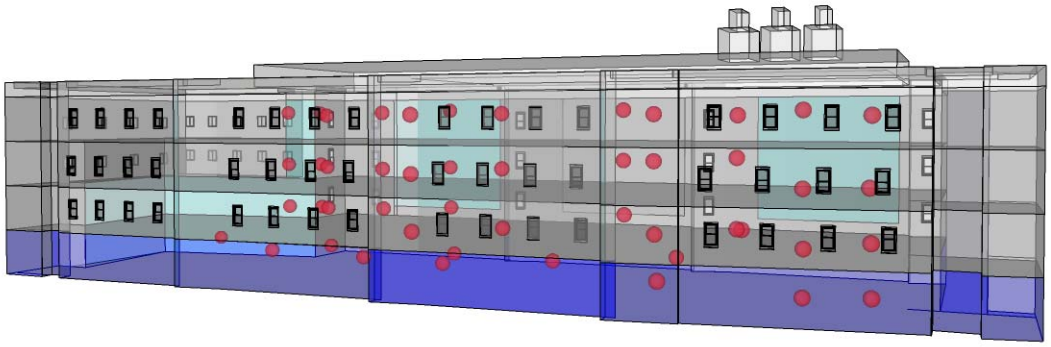

Figure 6.1: Locations of CORNET sensor nodes in ICTAS

6.3 Hardware and Software

This section will outline the hardware and software equipment used in deploying the fingerprinting system.

6.3.1 Hardware

As mentioned, the fundamental equipment for the fingerprinting system are the CORNET nodes. Each CORNET node has a USRP2 connected to a high performance server. The USRP2's designated as sensors on CORNET will sense users transmitting in the FRS frequency range of 462.5625 MHz to 467.7125 MHz. In comparison, the WDC nodes transmit

Table 6.1: WDCN communication specifications

Modulation	GMSK
Bit rate	100 Kbps
Transmitter (TX) amplitude	1.0
Transmitter (TX) gain	25
Bandwidth-time product	0.35
Packet size	1500 bytes

and receive at a center frequency of 500 MHz. Each CORNET node also is outfitted with a custom WBX daughterboard based on the Motorola RFIC4 [56]. The transceiver's frequency range is 100 MHz to 4 GHz with a bandwidth of up to 20 MHz.

Information about the communication links between the master and slave nodes are specified in Table 6.1. Downlinks and uplinks have the same physical layer specifications. Note that in our testbed, communication between the master and slave nodes is asynchronous. Fingerprint data or position estimates are transmitted only if available.

6.3.2 Software

Each CORNET node is connected to a Linux-based server with GNU Radio installed. GNU Radio signal processing components are used to establish the uplink and downlink channels specified in Table 6.1. Moreover, each slave node in the WDCN can implement either the DEDA or DBDA. These distributed localization algorithms are implemented as a series of Python scripts. The fingerprint radio map are saved as MATLAB .mat files to be accessed in Python.

To output the final estimated position calculated by the master server, PHP scripts are used to read a file with the user's trajectories. A screenshot of the web-based interface is shown in Figure 6.2. The red circles on the floor plan represent the approximate fingerprint location.

6.4 Functional Work and Data Flow

One of the crucial aspects of a WDCN is how tasks are disseminated and retrieved in an timely manner. Note that in order to minimize the makespan in estimating a position we determined each node's concerted effort in estimating a position. In other words, each slave server has a predetermined number of fingerprint positions (i.e. fixed AOR's) to analyze. First, we detail the functional work flow of the tasks and how it is accomplished. Then we look into the inner workings of our testbed's communication.

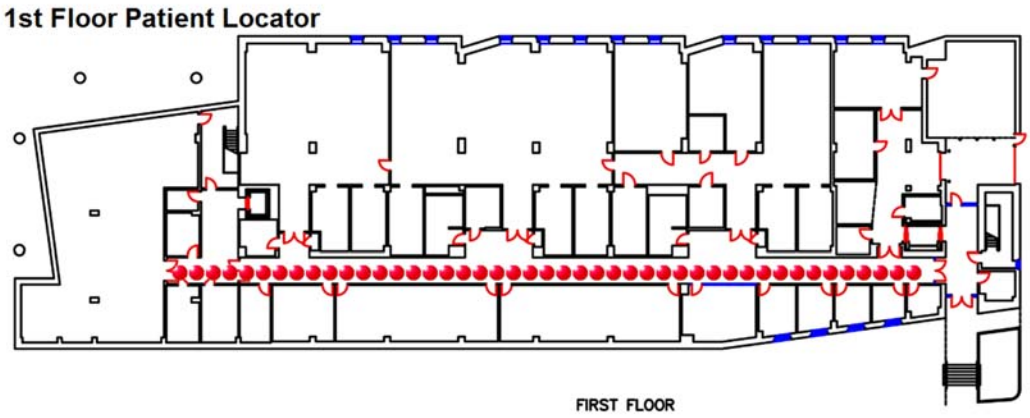

Figure 6.2: Screenshot of our web-based interface

6.4.1 Task Dissemination and Retrieval

Though any CORNET node can be assigned the role of a master server, we have specified the master-slave server relationships. The responsibility of distributing tasks and recovering the final estimated position belongs to our master server in the WDCN. In contrast, the slave servers cooperatively engage with the master server in completing the tasks. Each slave server is delegated by the master server an AOR, or a set of positions to check. Note that the term "client server" for our purposes will be used interchangeably with the term "slave server".

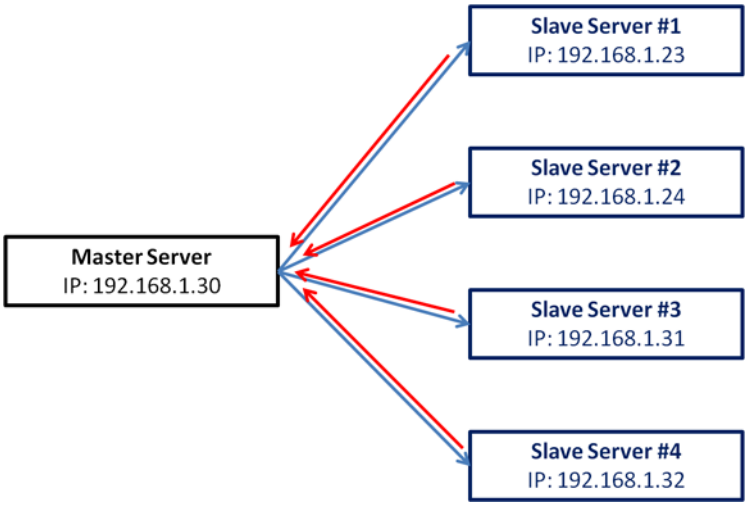

Figure 6.3: Block diagram of master server and slave server relationship

We now give a high level overview of how the WDCN component operates. Upon sensing a user transmitting, the CORNET nodes store the RSS values and send the current data via the CORNET Ethernet infrastructure to the master server. The master server will then collect and parse the RSS values to form a fingerprint. The current fingerprint is broadcast by the master server to all slave servers. Once the fingerprint is received, the slave servers compute their estimated position using one of the distributed localization algorithms for their AOR. A slave server's estimate is next sent back to the master server. Upon receiving all the estimated positions from the slave servers, the master server computes the true estimated position given the current estimates. Figure 6.4 shows the block diagram of task dissemination, and Figure 6.5 shows the events involved with task retrieval.

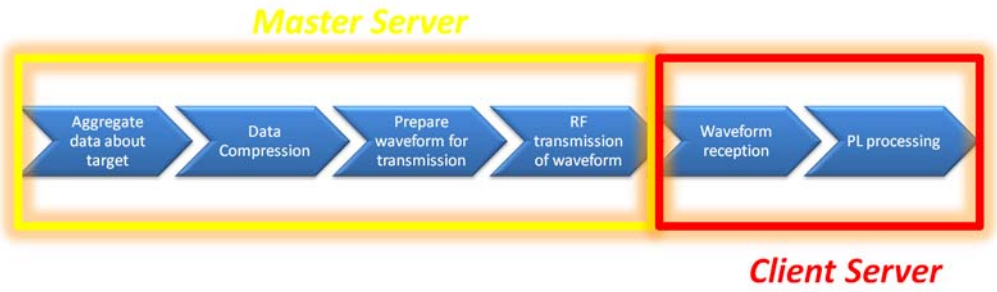

Figure 6.4: Phase I: Task dissemination

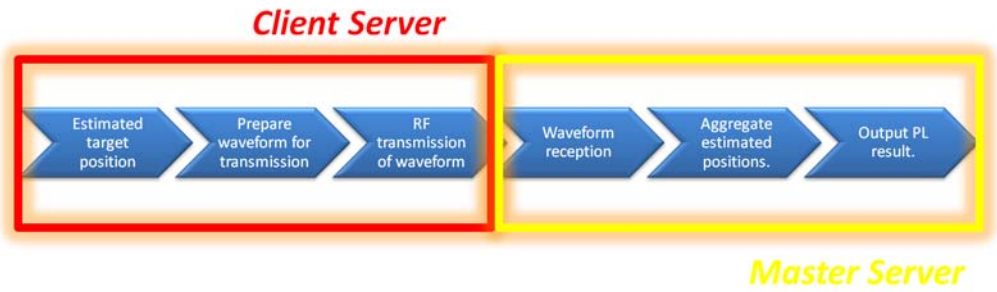

Figure 6.5: Phase II: Task retrieval

In the process of task dissemination in Figure 6.4, the master server aggregates the RSS measurements from the sensor nodes. Since CORNET has an Ethernet infrastructure, this aspect is easily accomplished. If the sensor nodes sent RSS values over-the-air (OTA), then the master server would need to implement a polling mechanism, time division multiple access (TDMA), or frequency division multiple access (FDMA). Since the WDCN is located on a separate floor, we chose not to implement an OTA sensor communication to minimize internode interference on CORNET. The next responsibility of the master server is to compress the amassed data to a usable format for the slave servers. This step concurrently

coincides with preparing the waveform for transmission. Once the waveform is transmitted and received by a slave server, it processes the data with one of the distributed localization algorithms. The step is simply referred to as position location (PL) processing.

The events of task retrieval in Figure 6.5 clearly only occurs after task dissemination. Here, the client server transmits its estimated position of the POI in a similar manner as the master server in Figure 6.4. Once the master server has each client server's estimated positions, then it will output the final estimated position of the POI.

6.4.2 WDCN Communication

The wireless medium access implemented for our WDCN is ALOHA-based, where the slave nodes do not "listen" to the the wireless channel before transmitting. In fact, the slave nodes will wait for a random amount of time prior to transmitting. If packets collide, then the master server will move to the most current fingerprint and automatically report that the user has not moved positions. We chose to implement an ALOHA-based MAC protocol because (1) we focused on meeting the goal of minimizing the makespan and (2) we have calibrated each slave server to send the packets of approximate equal length. Once the packet lengths are increased, the WDCN's MAC protocol must be reconfigured and benchmarked again.

The disadvantages of using ALOHA-based stem primarily from its lack of carrier sense multiple access (CSMA) with collision detection (CD) or collision avoidance (CA). With CSMA/CD or CSMA/CA, each ensures that the master server receives all estimated positions from the slave servers to output. Yet CSMA/CD could potentially increase the task's makespan if poor wireless channels are present because of retransmissions. Currently, there is no published research in developing a CSMA/CD-based or CSMA/CA-based MAC protocol for WDCN.

We also considered using GNU Radio's `tunnel.py`, which provides a minimal framework for a customizable medium access control. Datla, et. al have used this in [49]. This Python script uses the universal TUN/TAP virtual network drivers to tunnel packets from the USRP2 through Ethernet connections to the Linux kernel. What the user sees is a virtual TUN/TAP interface, often appearing as *gr0*. The major advantage of `tunnel.py` is the ability to control multiple USRP2's like a wired network. In turn, one could easily and dynamically delegate master-server relationships. Since `tunnel.py` uses error correction coding (ECC), the communication overhead exceeds the time taken to aggregate data and estimate a position. Furthermore, the process of task dissemination would occur sequentially and negate the purpose and benefits of WDC. Thus, we did not implement `tunnel.py` as a MAC.

6.5 Performance Evaluation

Prior the deployment of our fingerprint-based positioning system, approximately five hours were used to create the radio map offline. It contains fingerprints for 45 positions and is saved to a MATLAB .mat file named *radiomap_1f.mat*. Examples of the radio maps content are in Appendix E. Each location takes approximately one to two minutes to collect data, but setting up the equipment at each position could take up to five minutes. As stated earlier, twelve CORNET nodes, labeled N11 to N22, sensed the user's transmission.

We first tested the performance of the individual components, the distributed sensing system and WDC system, of our fingerprinting separately. To test the first component, the distributed sensing system we moved to each location while facing a northwest direction and sampled the data 10 times at a particular location. Each stored sample is then used to determine its Euclidean distance with the individual fingerprints in the radio map. It is here that the WDCN is tested. We conducted two system tests with the same radio map. The results indicate that the RSS's time dependency did not create problems for the positioning system.

Figure 6.6 and Figure 6.7 are screenshots of our WDCN estimating the POI's location. The window screen on the left side in both screenshots is the designated master server, while the four smaller screens output the results of the slave servers. In some of the slave server screens, one can see their estimate of the POI's location. Recall that in both distributed algorithms that the master server will aggregate all these estimates and finalize the location solution. The final estimated fingerprint location of the POI is indicated with the yellow rectangle. For this particular iteration, both distributed algorithms estimated that the POI is in Box 2, or fingerprint location 2. Without a graphical rendering of the estimated solution, the alphanumeric numbering is not meaningful for the user.

The precision of our positioning system is approximately 75% at approximately 3.048 m (10'). The average makespan for estimating a single position, not including the communication time, for both distributed localization algorithms is in Table 6.2. Note the total time includes the processing time of all client servers and the master server. During the online phase, DEDA, in general, performed more quickly than DBMA by 1 second.

Table 6.2: Computational time of distributed localization algorithms

Positioning Algorithm	**Single Client Server**	**Total Time**
Distributed Euclidean Distance	1 seconds	3 seconds
Distributed Bayesian Modeling	2 seconds	4 seconds

As highlighted in Chapter 3, WDC can reduce per-node and network power, energy, and processing resource requirements. For our WDC-based fingerprinting position system, we consider the total computational complexity as a direct contributor of each slave server's energy consumption. Given R training patterns for each distributed localization algorithm,

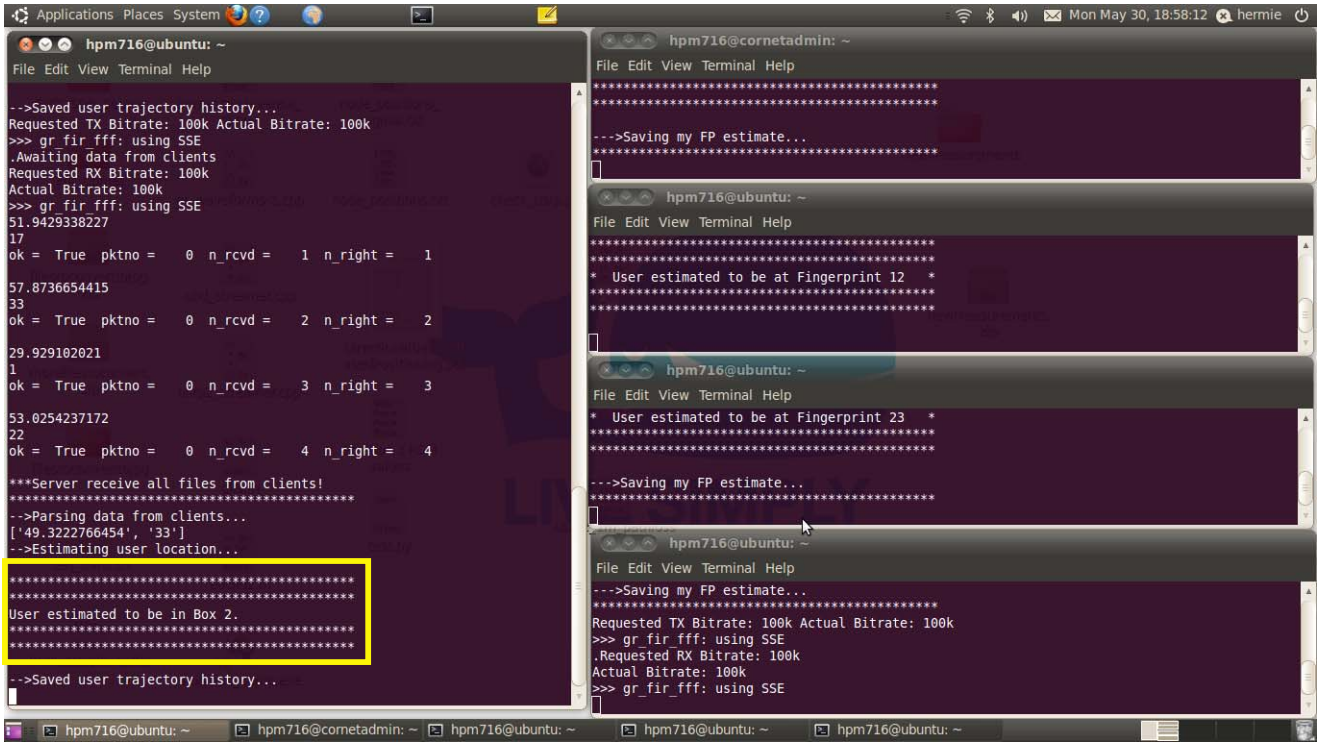

Figure 6.6: Screenshot of WDCN using DEDA

we can estimate the complexity requirements for the online phase of our system. We assume that there is a total of n fingerprint locations and N sensor nodes used.

The total computational complexity of DEDA is $\mathcal{O}(n)$, where n is the number of fingerprint locations. Then we can conclude that the average computational complexity experienced by each client server is $\mathcal{O}(n/p)$, where p is the number of AOR's assigned by the master server. In our system, the average computational complexity is $\mathcal{O}(n/4)$. For DBMA, the total computational complexity is $\mathcal{O}(n)$. Then the average computational complexity for each client server is also $\mathcal{O}(n/4)$. However, DBMA has a search time per client server of $\mathcal{O}\left(n/4\left(\log u+1\right)\right)$, where u is the average number of unique patterns at each fingerprint location. These results are summarized in Table 6.3 and Table 6.4.

Table 6.3: Total computational complexities of distributed localization algorithms during the online phase

Algorithm	**Computation**	**Searching**	**Sorting**
Distributed Euclidean Distance	$\mathcal{O}(n)$	N/A	$\mathcal{O}(n\log n)$
Distributed Bayesian Modeling	$\mathcal{O}(n)$	$\mathcal{O}\left(n\left(\log u+1\right)\right)$	$\mathcal{O}(n\log n)$

In general, DEDA requires a memory space of $n\times(N+d)$, where d is the units (e.g bytes) of the radiomap's dimensions, and a search space of n. The DBMA demands $n\times u(N+$

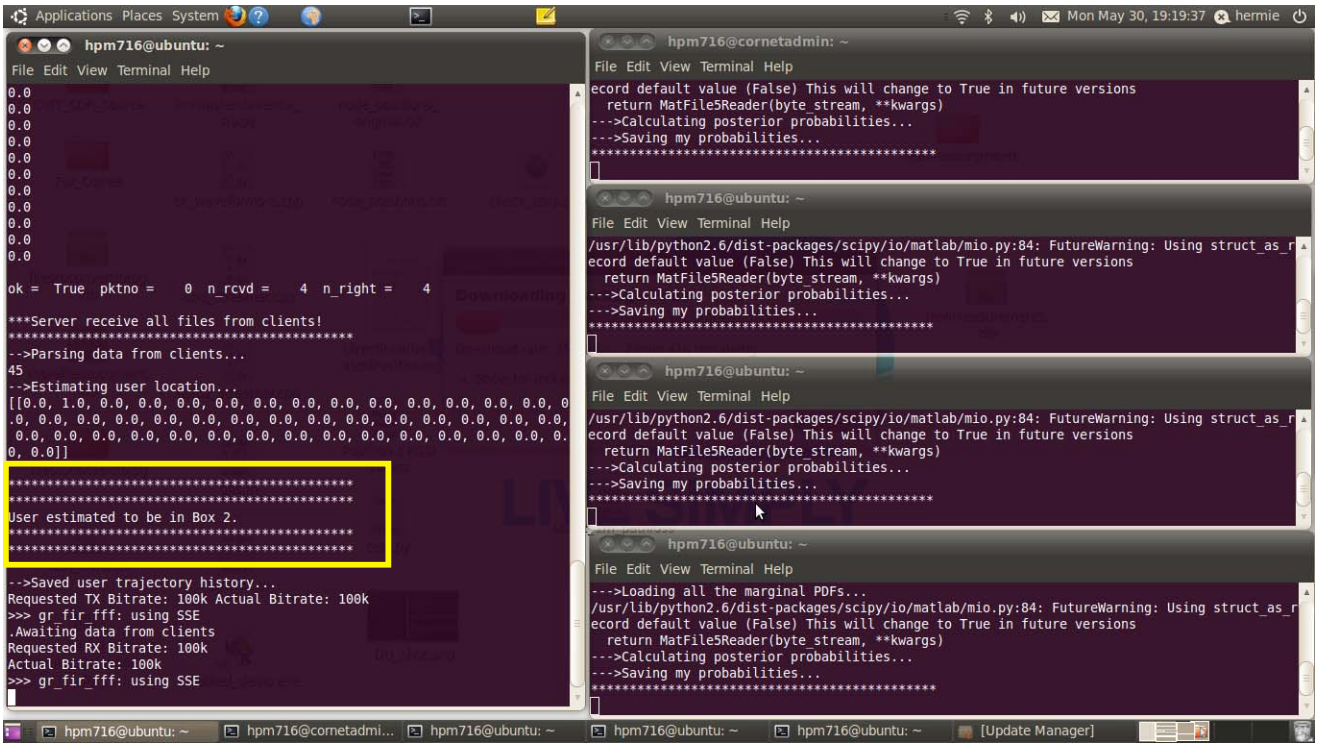

Figure 6.7: Screenshot of WDCN using DBMA

Table 6.4: Computational complexities of distributed localization algorithms during the online phase for a single slave node

Algorithm	**Computation**	**Searching**	**Sorting**
Distributed Euclidean Distance	$\mathcal{O}(n/4)$	N/A	$\mathcal{O}(n/4 \log n/4)$
Distributed Bayesian Modeling	$\mathcal{O}(n/4)$	$\mathcal{O}\left(n/4\left(\log u+1\right)\right)$	$\mathcal{O}(n/4 \log n/4)$

$d + 1$) of memory with a search space of $n \times u$. Therefore, DEDA has a shorter search and computational time than DBMA. Since DBMA generally requires a large number of training data, its offline phase will take a significant amount of time. This laborious offline phase cannot be reduced using WDC. Instead, WDC enables a node to only store and process a minimal or limited fingerprint database. This directly translates to resource savings. Since the WDC nodes we used are not necessarily resource constrained, the benefits of the reduced computational complexities in Table 6.4 would be most prominent in a mobile device with limited resource and computational capabilities.

6.6 Summary

This chapter is the most crucial contribution for establishing a WDC-based fingerprinting position system in ICTAS. What must be considered first is the time-consuming offline data collection to create the radio map. This radio map is a fundamental input into each distributed localization algorithm, so it must be carefully completed. In regards to the WDCN, position calculations work effectively in a distributed manner. However, using the ALOHA-based MAC protocol significantly increased the makespan of the system.

Chapter 7

Conclusions and Future Work

Current literature in fingerprint-based positioning systems often focus on empirical results and performance studies of locating a user with known transmitter-receiver relationship. In general, these systems accomplish this by relying on centralized computations on a resource constrained device or central server. What this thesis has developed are four distributed target localization algorithms for a fingerprint-based indoor positioning system. These algorithms can be implemented in a DC or WDC environment without significant changes to current communications infrastructure. Using one of the distributed algorithms enables the system designer to optimize the network's computing resources for both responsiveness and economy. With the increasing need of location-based services and overall dependence on mobile devices, this type of joint optimization is extremely important in wireless communications.

How to distribute localization algorithms was investigated extensively in Chapter 4. One method of distributing the position calculations is not applicable for the WDC environment. What we recommend is that the search area be partitioned into areas of responsibilities (AOR) to be monitored by a group of WDC nodes. These nodes are then only responsible for calculating an estimated position for their AOR. In turn, the WDC node's search area is constrained and quickens the fingerprint search process. With several nodes simultaneously searching their AOR, the makespan of the search process is significantly reduced.

Prior to implementing a prototype fingerprint-based positioning system, exhaustive measurements were taken of the indoor wireless channel in Chapter 5. What we found was that the RSS values in ICTAS is a non-stationary process. We analyzed this and other properties of the RSS in order develop a mathematically tractable set of assumptions. These assumptions then are applied in testing the four distributed localization algorithms. The distributed euclidean distance algorithm (DEDA) and the generalized regression neural network (GRNN) performed the best. A location accuracy of 3.048 m (10') is found for 74% of the empirical data using DEDA. This accuracy is comparable to GRNN's location accuracy of 3.048 m (10') of 71%. We then apply two of the distributed localization algorithms, DEDA and

distributed Bayesian modeling algorithm (DBMA), to a prototype WDC-based fingerprint positioning system in Chapter 6.

7.1 Contributions

This section lists the major contributions of this thesis:

1. Extensive RSS measurement analysis confirmed several properties of RSS. Though the harsh indoor wireless environment significantly corrupts RSS, we found ways to mitigate RSS's randomness and proactively use it to benefit location determination. Most notably, we found that enlargening the separation between positions and increasing the number of sensors can benefit the system's accuracy and precision.

2. We proposed four distributed localization algorithms that can be applied to a DC and WDC environment. Each algorithm can be used in large indoor service areas as a GPS alternative positioning system.

3. We developed a configurable and scalable prototype WDC-based fingerprint position system in Virginia Tech's ICTAS research facility. We used the prototype to showcase and validate two distributed localization algorithms.

7.2 Future Work

A significant amount of work has been done in regards to the setup of our fingerprint positioning system and evaluating our distributed localization algorithms. Therefore, this research in this thesis provided the foundations of future studies in fingerprinting in ICTAS and creating a WDCN using CORNET nodes. However, there exists two major areas for possible improvement to our research. The first area deals with the system's scalability of the fingerprint system. The second area focuses on further studying how can conventional positioning techniques be calculated in distributed manner.

7.2.1 Scalability of the WDC-based Positioning System

The current fingerprint positioning system is limited to localizing users on the first floor of ICTAS. We only use a subset of nodes on the second floor for the WDCN. However, we could substitute the WDCN nodes with other mobile devices (i.e. a laptop computer with a separate USRP2 device or even an Android smartphone). Then the distributed sensing system could be implemented across all four floors of ICTAS. What this effectively does is

to enable mobile and more user-friendly location services. It could completely replace the web-based interface with standalone software.

Scaling the positioning system though introduces new challenges. The most significant challenge would be how to aggregate the RSS values from up to 48 nodes in an efficient manner to be computed. Because the system designer would be most likely interested in a deterministic sampling frequency, multi-hop protocols like Trickle [64] could be implemented. A related challenge would be how to communicate between the distributed sensing system and the WDCN. A more complex positioning system would implement the WDCN with the existing distributed sensing architecture. The current positioning system has no major issues communicating between the distributed sensing components and WDCN because we leverage CORNET's Ethernet structure. Expanding the service area in ICTAS will also require more efficient sensing routines on each CORNET node to better track users.

7.2.2 Improving Distributed Localization Algorithms

Conventional position techniques like TOA and TDOA often employ a least squares algorithm as mentioned in Section 2.1. We highlight in Section 4.4 one method on how to parallelize solving a system of equations. But for a WDCN, the method is highly communicative. Another promising method is to use multisplitting [65]. The general idea of multisplitting is to replace a large-scale linear (or nonlinear) problem by a set of subproblems, which can each be solved locally and independently in parallel. In the case of a LS optimization, we seek to minimize the the sum of the squares of the errors or

$$\min_x \|Ax - b\|^2 \tag{7.1}$$

To multisplit an LS optimization, first, let A be partitioned such that $A = (A_1, A_2, \ldots, A_p)$, $A_j \in \mathbb{R}^{mxn_j}$, and $\sum_{j=1}^{p} n_j = n$. Now define

$$f(x) = \left\| \sum_{j=1}^{p} A_j X_j - b \right\|^2 \tag{7.2}$$

and

$$b_j(x) = b - \sum_{i \neq j}^{p} A_i X_i, \quad 1 \leq j \leq p \tag{7.3}$$

where $x = (X_1, X_2, \ldots, X_p)^T$. Take positive scalars α_j^k such that $\sum_{j=1}^{p} \alpha_j^k = 1$ and the Least Squares Multisplitting (LSMS) is defined by the iteration

$$x^{k+1} = \sum_{j=1}^{p} \alpha_j^{k+1} Z^{j,k+1} \tag{7.4}$$

where $Z^{j,k}$ is x^k with update X_j in the j^{th} position and X_j is the solution to the subproblem

$$\min_{X_j \in \mathcal{R}^{n_j}} \left\| A_j X_j - b_j(x^k) \right\|^2, \quad 1 \leq j \leq p. \tag{7.5}$$

Using multisplitting optimization techniques would be highly applicable for localization in an outdoor environment and tactical scenarios, so long as the solutions to the subproblems converge.

Bibliography

[1] E. G. R. Taylor. *The haven-finding art: a history of navigation from Odysseus to Captain Cook*. American Elsevier Publishing Company, INC, New York, 1971.

[2] N. Bowditch. *New American Practical Navigator*. Bethesda: National Imagery and Mapping Agency, 1995.

[3] William Edward May. *A History of Marine Navigation*. G. T. Foulis & Co. Ltd., Oxfordshire, 1973.

[4] J. H. Reed, K. Krizman, B. Woerner, T. Rappaport. An overview of the challenges and progress in meeting the E-911 requirement for location services. *IEEE Communications Magazine*, vol. 36, April 1998.

[5] D. Datla, S. Raghunandan, X. Chen, S. Bera, H. Mendoza, S.M. Shajedul Hasan, J.H. Reed. Distributed Computing for Collaborative Software Radio. Blacksburg, VA: 2010.

[6] D. Niculescu and B. Nath. "Ad hoc positioning system (APS) using AOA." *IEEE INFOCOM*, April 2003.

[7] A. Nasipuri and K. Li. "A directionality based location discovery scheme for wireless sensor networks." *First ACM International Workshop on Wireless Sensor Networks and Applications*. Atlanta, GA, Sept. 2002.

[8] T. Rappaport. *Wireless Communications: Principle and Practice*. Prentice Hall, 1996.

[9] W.C. Jakes. *Microwave Mobile Communications*. IEEE Press, 1994.

[10] A. J. Coulson, A. G. Williamson, R. G. Vaughan. "A statiscal basis for lognormal fading effects in multipath fading channel." *IEEE Transactions on Communications*, vol. 46, April 1998.

[11] T. Logsdon. *The Navstar Global Positioning System*. Van Nostrand Reinhold, 1999.

[12] J. Hightower, R. Want, and G. Borriello. "SpotON: An indoor 3D location sensing technology based on RF signal strength." UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA, Feb. 2000.

[13] K. Langendoen and N. Reijers. "Distributed localization in wireless sensor networks: a quantitative comparison." *Computer Networks*, 43(4): 499-518, 2003.

[14] A. Savvides, H. Park, M. Srivastava. "The bits and flops of the N-hop multilateration primitive for node localization problems." First ACM International Workshop on Wireless Sensor Networks and Application, Atlanta, GA, 2002.

[15] C. Savarese, K. Langendoen, and J. Rabaey. "Robust positioning algorithms for distributed ad-hoc wireless sensor network." USENIX Technical Annual Conference, Monterey, CA, 2002.

[16] N. Bulusu, J. Heidenmann, and D. Estrin. "GPS-less low cost outdoor localization for very small devices." *IEEE Personal Communications Magazine*, 7(5): 28-34, Oct. 2000.

[17] N. Bulusu, D. Estrin, L. Girod, and J. Heidenmann. "Scalable coordination for wireless sensor networks: Self-configuring localization systems." *Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA '01)*. July 2001.

[18] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. "Range-free localization schemes for large scale sensor networks." *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, 81-95. ACM Press, 2003.

[19] J. Hightower, R. Want, and G. Borriello. "Location systems for ubiquitious computing." *Computer*, 34(8): 57-66, 2001.

[20] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. "Multi-camera multi-pereson tracking for easyliving." *Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS 2000)*, 3-10. IEEE Computer Society, 2000.

[21] M. J. Feuerstein. Urban and Indoor Location using Pattern Matching of Wireless Network Measurements." Invited Workshop on Opportunistic RF Localization for Next Generation Wireless Devices, Worcester Polytechnic Institute, June 2008.

[22] A. Savvides, C. C. Han, and M. B. Srivastava. "Dynamic fine-grained localization in ad-hoc networks of sensors." *Proceedings of MobiCom*, 166-179, 2001.

[23] P. Bahl and V. N. Padmanabhan. "RADAR: An In-Building RF-based User Location and Tracking System." *Proceedings of the IEEE INFOCOM 2000*, vol. 2, 775-784, Mar. 2000.

[24] A. M. Ladd, K. E. Bekris, A. Rudys, L. E. Kavraki, and D. S. Wallach. "Robotics-based location sensing using wireless Ethernet." *Wireless Networks*, vol. 11, no. 1-2, 180-204, 2005.

[25] U. Grossmann, M. Schauch, and S. Hakobyan. "RSSI based WLAN indoor positioning with personal digital assistants." *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Sept. 2007.

[26] M. Meurer, S. Heilmann, D. Reddy, T. Weber, and P. W. Baier. "A signature-based localization technique relying on covariance matrices of channel impulse responses." *Proceedings of the 2nd Workshop on Positioning, Navigation, and Comunication (WPNC '05) and 1st Ultra-Wideband Expert Talk (UET '05)*, 31-40, 2005.

[27] K. Kaemarungsi and P. Krishnamurthy. "Modeling of the Indoor Positioning Systems Based on Location Fingerprinting." *Proceedings of IEEE INFOCOM*, 2004.

[28] P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis. "On indoor position location with wireless LANs." Telecommunications Program, University of Pittsburgh, PA, Tech. Rep., 2002.

[29] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz. "A Probabilistic Room Location Service for Wireless Networked Environments." *Proceedings of Ubiquitous Computing*, 18-34, 2001.

[30] M. Brunato and R. Battiti. "Statistical Learning Theory for Location Fingerprinting in Wireless LANS." *Computer Networks*, vol. 47, no. 6, 825-845, 2005.

[31] D. Datla, X. Chen, T. Tsou, S. Raghunandan, S. M. Shajedul Hasan, J. H. Reed, B. Feete, C. B. Dietrich, J. Kim, and T. Bose. "Wireless Distributed Computing: A Survey of Research Challenges." *IEEE Communications*, 2011.

[32] D. Datla, T. Tsou, T. R. Newman, J. H. Reed, and T. Bose. "Waveform level computational energy management in software defined radios." *SDR '09 Technical Conference and Product Exposition*, Dec. 2009.

[33] N. Santoro. *Design and Analysis of Distributed Algorithms*. Wiley-Interscience, Oct. 2006.

[34] D. Datla. X. Chen. T. R. Newman, J. H. Reed, and T. Bose. "Power efficiency in wireless network distributed computing." *IEEE Vehicular Technology Conference*, Sep. 2009.

[35] G. Tel. *Topics in distributed algorithms*, vol. 1, 1991.

[36] N. Lynch. *Distributed Algorithms*, 1st ed., Mar. 1996.

[37] J. Mitola, III, "Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio." Doctor of Theology, Royal Institute of Technology (KTH), Stockholm, 2000.

[38] H. Celebi and H. Arslan. " Cognitive positioning systems." *IEEE Transactions on Wireless Communications*, vol. 6, no. 12, 4475-4483, 2007.

[39] H. Celebi and H. Arslan. " Utilization of location information in cognitive wireless network." *IEEE Wireless Communications*, vol. 14, no. 4, 6-13, 2007.

[40] S. Yarkan and H. Arslan, "Exploiting location awareness towards improved wireless ystem design in cognitive radio." *IEEE Communications Magazine*, vol. 46, no. 1, 123-136, Jan. 2008.

[41] A. Saleh and R. Valenzuela. "A Statistical Model for Indoor Multipath Propagation." *IEEE Journal on Selected Areas in Communication*, vol. JSAC-5, no. 2, 128-137, Feb. 1987.

[42] J. Caffrey Jr. and G. Stuber. "Overview of Radiolocation in CDMA Cellular Systems." *IEEE Communications Magazine*, Apr. 1998.

[43] K. Pahlavan and P. Krishnamurthy. *Principles of Wireless Networks: A Unified Approach*, Prentice Hall PTR, 2002.

[44] K. Pahlavan, P. Krishnamurthy, and J. Beneat. "Wideband radio propagation modeling for indoor geolocation applications". *IEEE Communications Magazine*, 60-65, Apr. 1998.

[45] N. Patwari, A. O. Hero III, J. Ash, R. I. Moses, S. Kyperountas, and N. S. Correal. "Locating the nodes - cooperative localization in wireless sensor networks." *IEEE Signal Processing Magazine*, vol. 22, no. 4, 54-69, July 2005.

[46] J. A. Costa, N. Patwari, and A. O. Hero III. "Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks." *ACM Transactions on Sensor Networks*, vol. 5, Jun. 2005.

[47] F. Reichenbach, A. Born, D. Timmermann, and R. Bill. "A Distributed Linear Least Suqres Method for Precise Localization for Low Complexity in Wireless Sensor Networks." *Proceedings of the 2nd International Conference on Distributed Computing in Sensor Systems (DCOSS '06)*, 2006.

[48] M. Liggins II, C. Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos. "Distributed Fusion Architectures and Algorithms for Target Tracking." *Proceedings of IEEE*, vol. 85, no. 1, 95-107, Jan. 1997.

[49] D. Datla, S. Raghunandan, X. Chen, S. Bera, H. Mendoza, S. M. Hasan, and J. H. Reed. "Distributed Computing for Colloborative Software Radio." ONR Grant No. N30014-07-01-0536, Sept. 2010.

[50] R. Battiti, et al. "Neural Network Model for intelligent networks: deriving the location from signal patterns." *The First Annual Symposium on Autonomous Intelligent Networks and Systems*, 2002.

[51] R. Battiti, T. L. Nhat, and A. Villani. "Location-Aware Computing: A Neural Network Model for Determining Location in Wireless LANS." University of Trento Technical Report DIT-02-0083, 2002.

[52] L. L. Scharf. *Statistical Signal Processing – Detection, Estimation, and Times Series Analysis*, 2nd ed., Addison-Wesley, 1991.

[53] D. J. Torrieri. "Statistical theory of passive location systems." *IEEE Transaction on Aerospace and Electronic Systems*, AES-20 (2), 183-198, 1984.

[54] Ettus. "USRP2 Product Data Sheet," Online, May 2011. [Online]. Available: `http://www.ettus.com/downloads/ettus_ds_usrp2_v5.pdf`

[55] Motorola. "Talkabout MR350R," Online, May 2011. [Online]. Available: `http://www.motorola.com/Business/US-EN/Business+Product+and+Services/Two-Way+Radios+-+Consumers/MR350R_Talkabout_Two-Way-Radio_US-EN`

[56] "CORNET," Online, May 2011. [Online]. Available: `http://www.cornet.wireless.vt.edu/trac/wiki/CORNET`

[57] GNU Radio. "GNU Radio," Online, May 2011. [Online]. Available: `http://gnuradio.org/redmine/wiki/gnuradio`

[58] B. Sklar. "Rayleigh fading channels in mobile digital communication systems: I. Characterization. *IEEE Communications Magazine*, vol. 35, no. 7, 90-100, July 1997.

[59] B. G. Tabachnick and L. S. Fidell. *Using Multivariate Statistics*, 3rd ed. New York, NY: Harper Collins, Mar. 1996.

[60] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. E. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Taber, P. Powledge, G. Borriello, and B. N. Schilit. "Place Lab: Device Positioning Using Radio Beacons in the Wild." *Proceedings of International Conference on Pervasive Computing*, 116-133, 2005.

[61] R. Want, A. Hopper, V. Falcao, and J. Gibbons. "The Active Badge Location System." *ACM Transactions on Information Systems*, vol. 40, no. 1, 91-102, Jan. 1992.

[62] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. "The Cricket Location Support System." *Proceedings of ACM International Conference on Mobile Computing and Networking (MOBICOM '00)*, Boston, MA, 32-43, Aug. 2000.

[63] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. "LANMARC: Indoor Location Sensing using Active RFID." *Wireless Networks*, vol. 10, no. 6, 701-710, Nov. 2004.

[64] P. Levis, N. Patel, S. Shenker, and D. Culler. "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks." Technical report, University of California at Berkeley, 2004.

[65] R. A. Renaut. "A parallel multisplitting solution of the least squares problem." *Numerical Linear Algebra with Applications*, 4, 1-21, 1997.

Appendix A

Histograms of a 22 Fingerprint Database

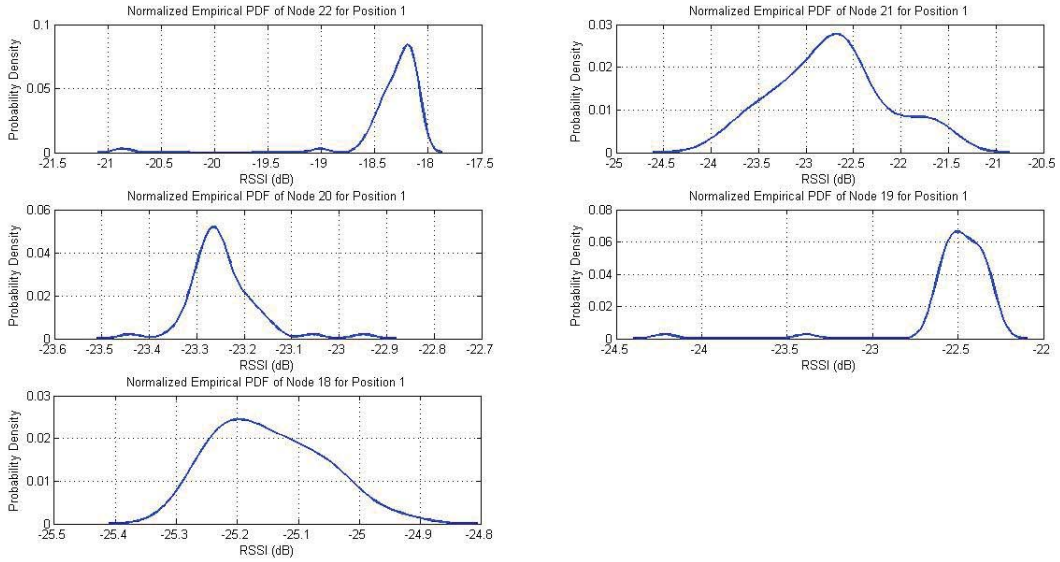

Figure A.1: Histogram of Position 1

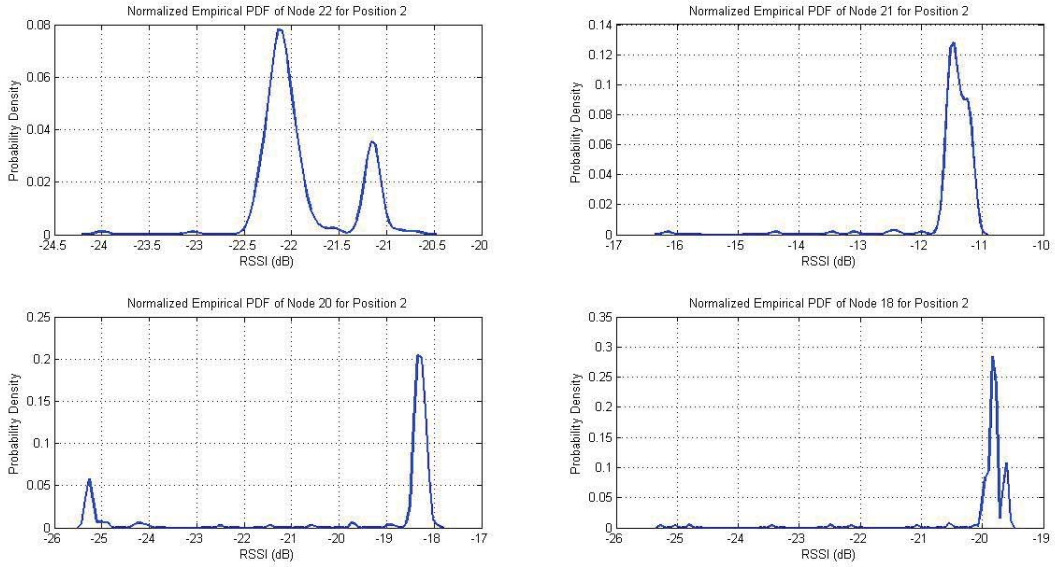

Figure A.2: Histogram of Position 2

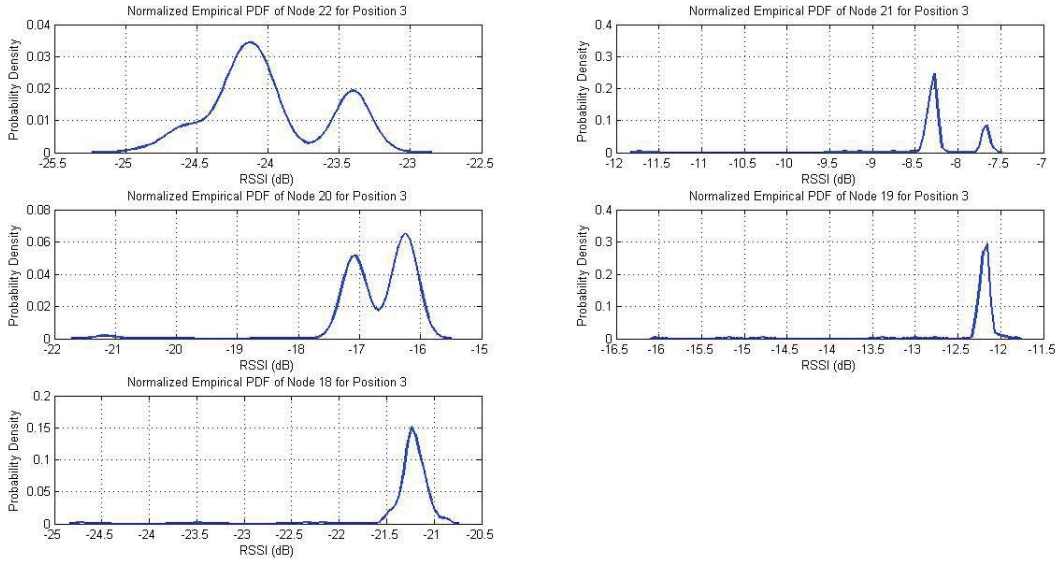

Figure A.3: Histogram of Position 3

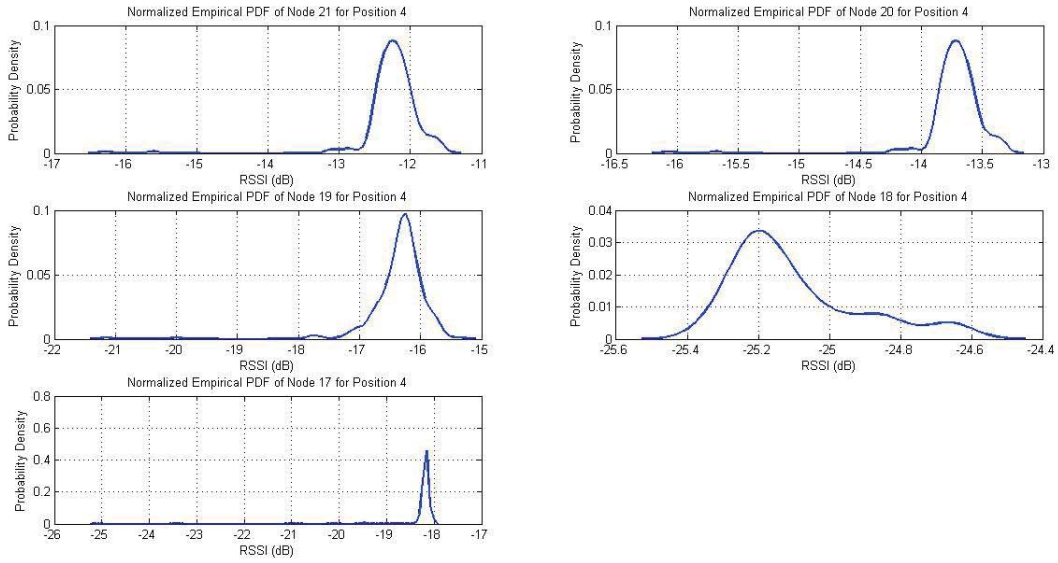

Figure A.4: Histogram of Position 4

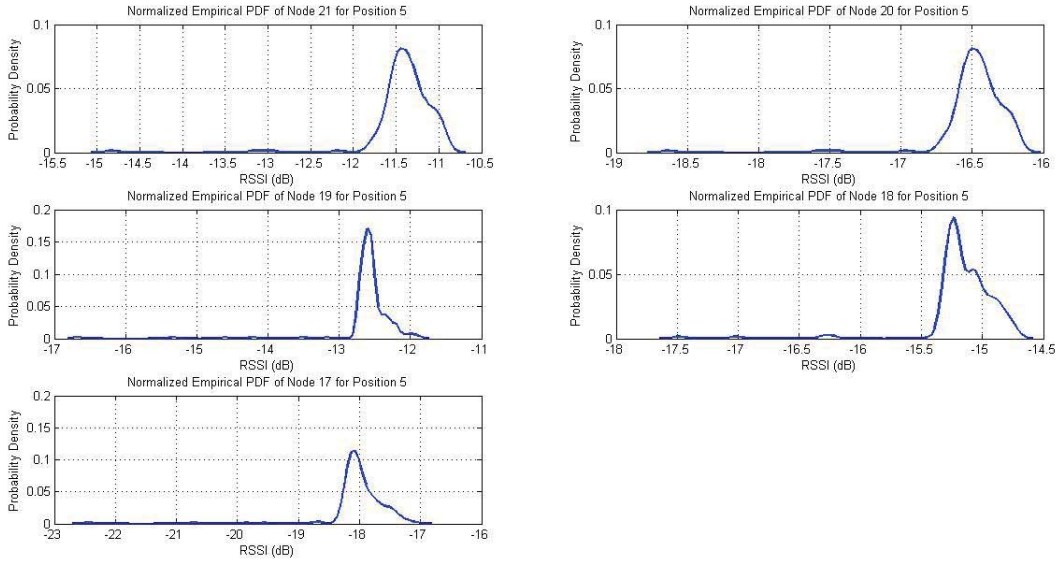

Figure A.5: Histogram of Position 5

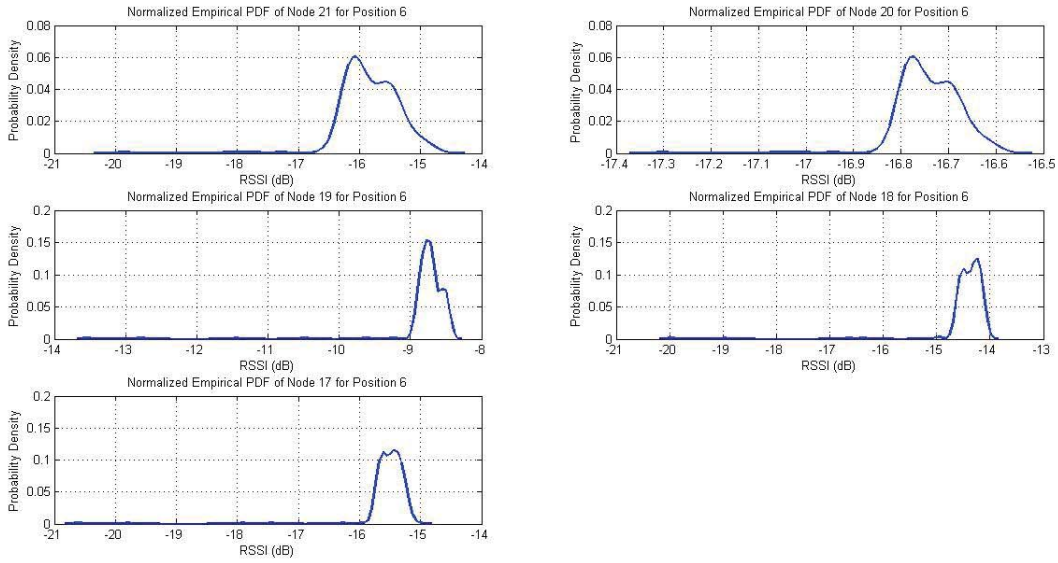

Figure A.6: Histogram of Position 6

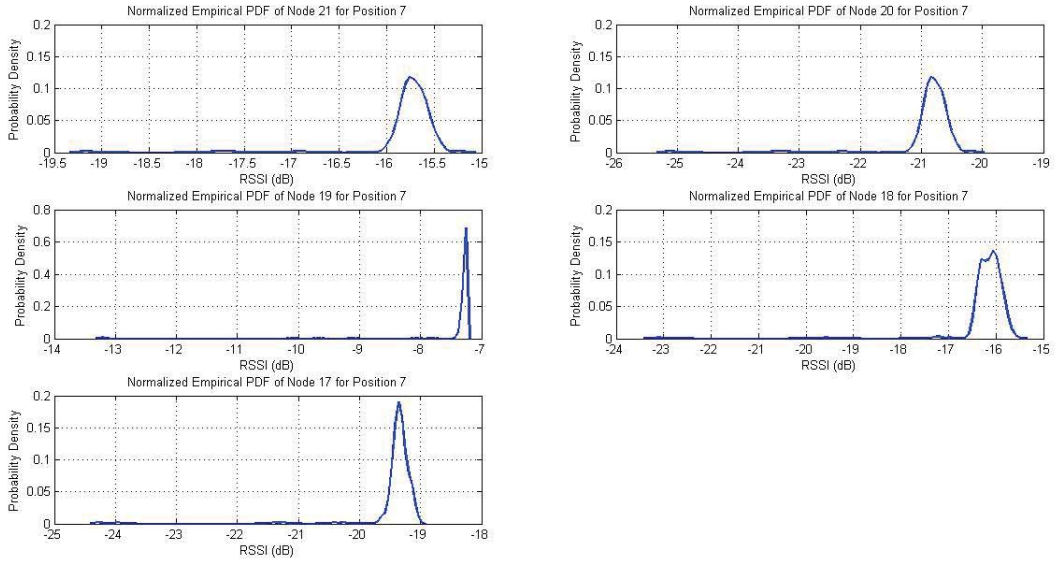

Figure A.7: Histogram of Position 7

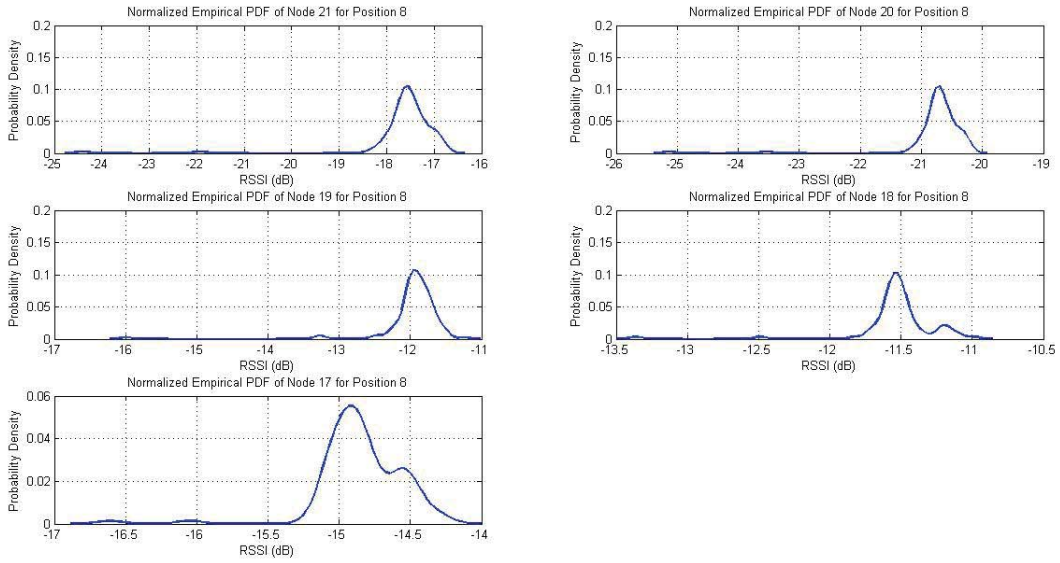

Figure A.8: Histogram of Position 8

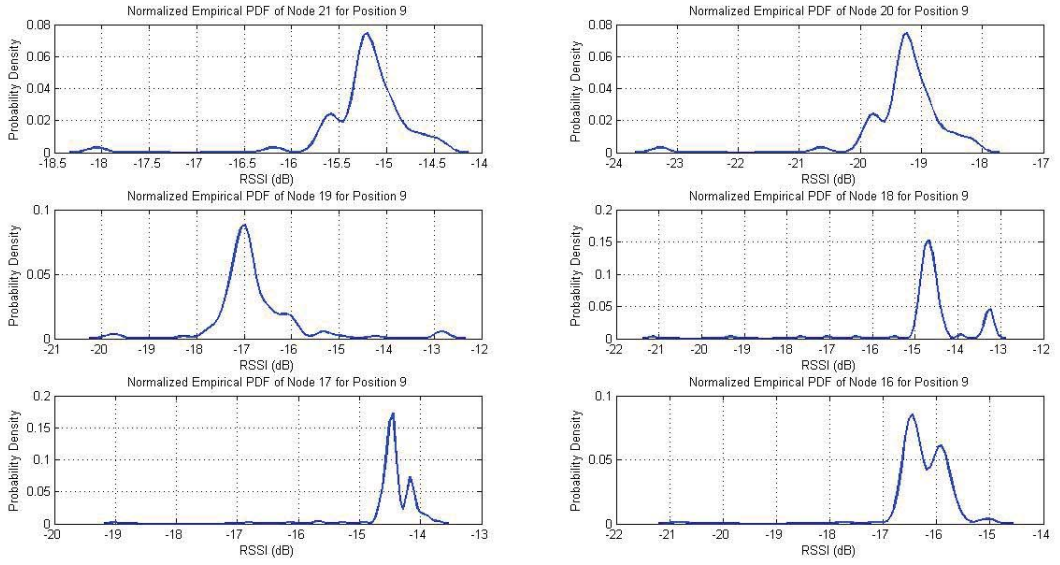

Figure A.9: Histogram of Position 9

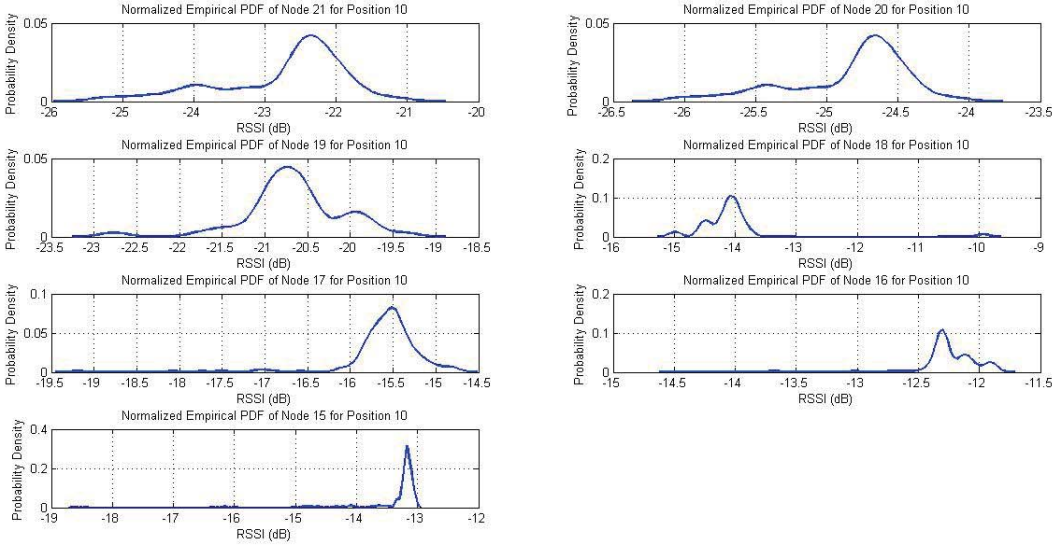

Figure A.10: Histogram of Position 10

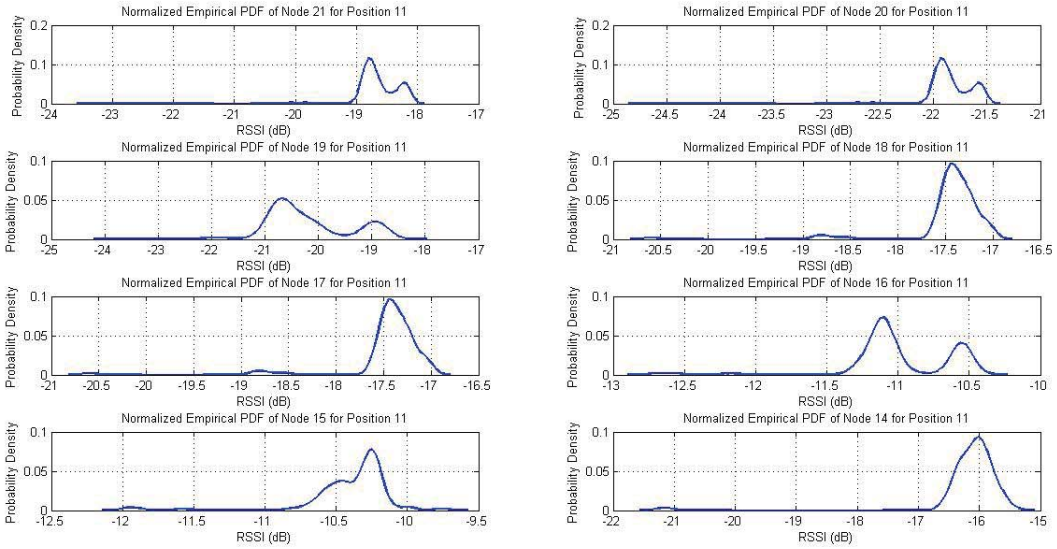

Figure A.11: Histogram of Position 11

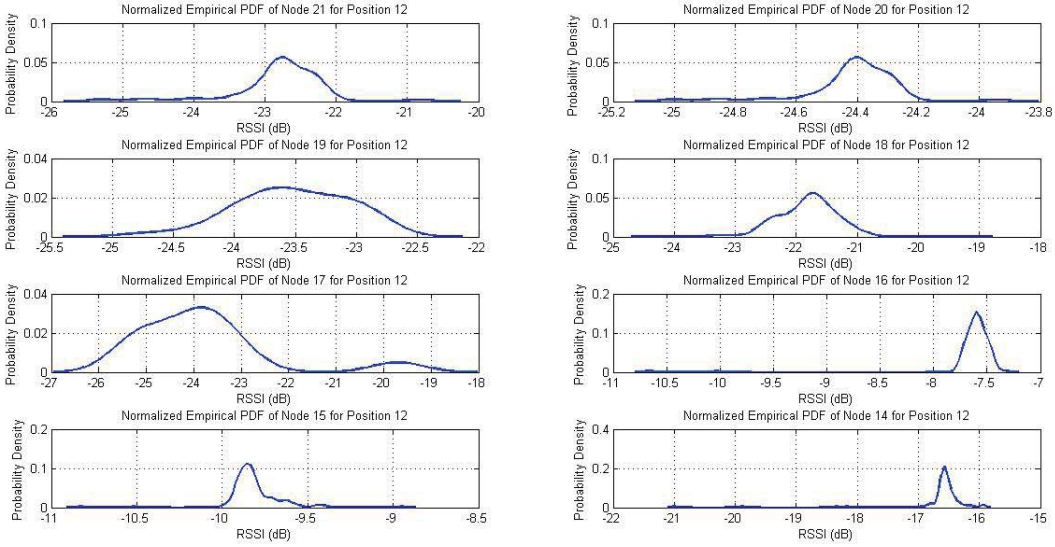

Figure A.12: Histogram of Position 12

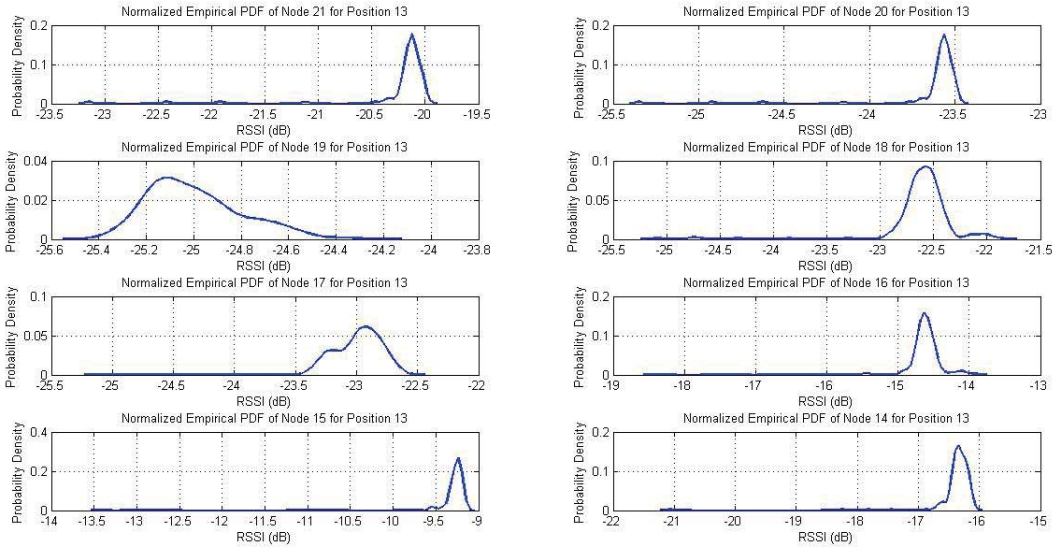

Figure A.13: Histogram of Position 13

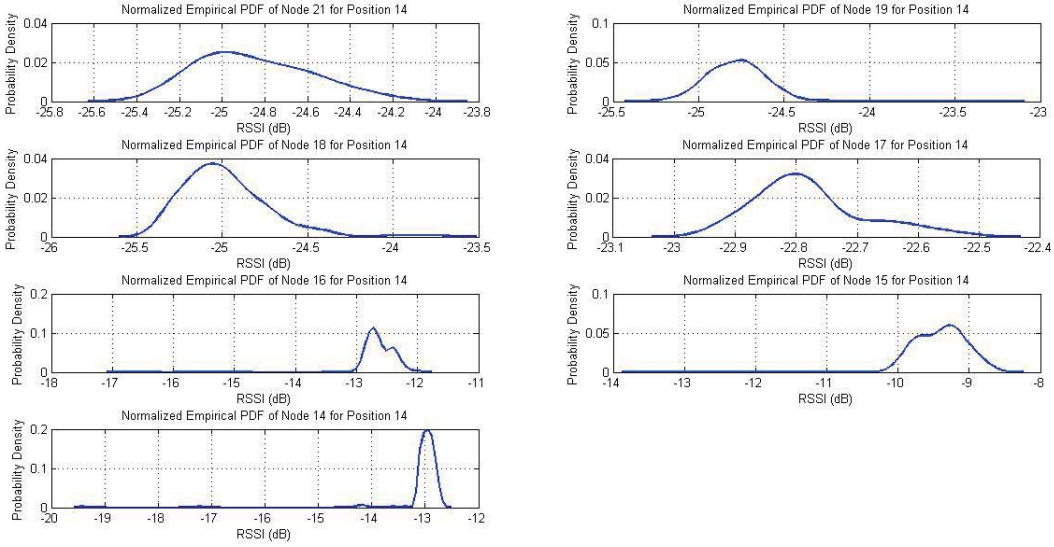

Figure A.14: Histogram of Position 14

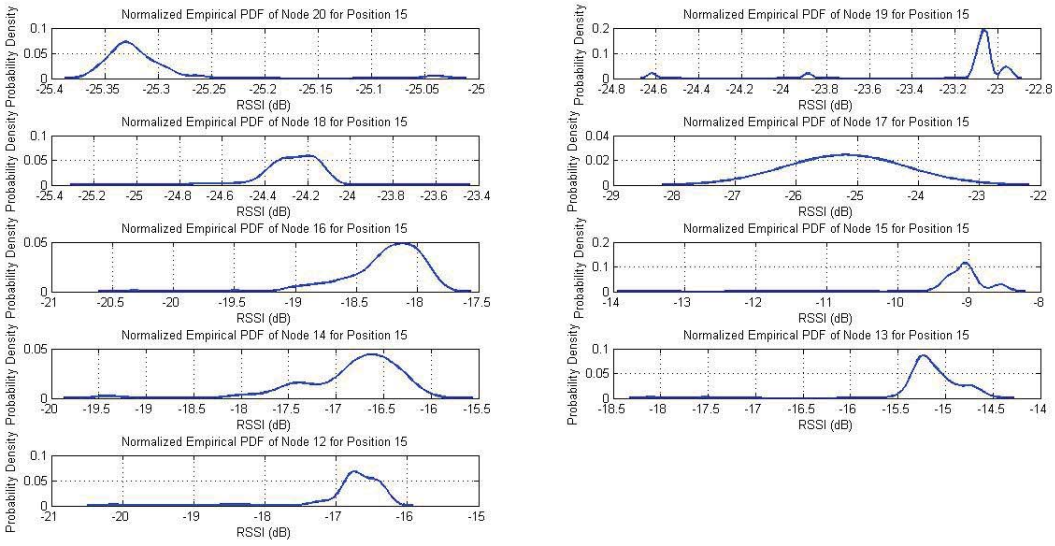

Figure A.15: Histogram of Position 15

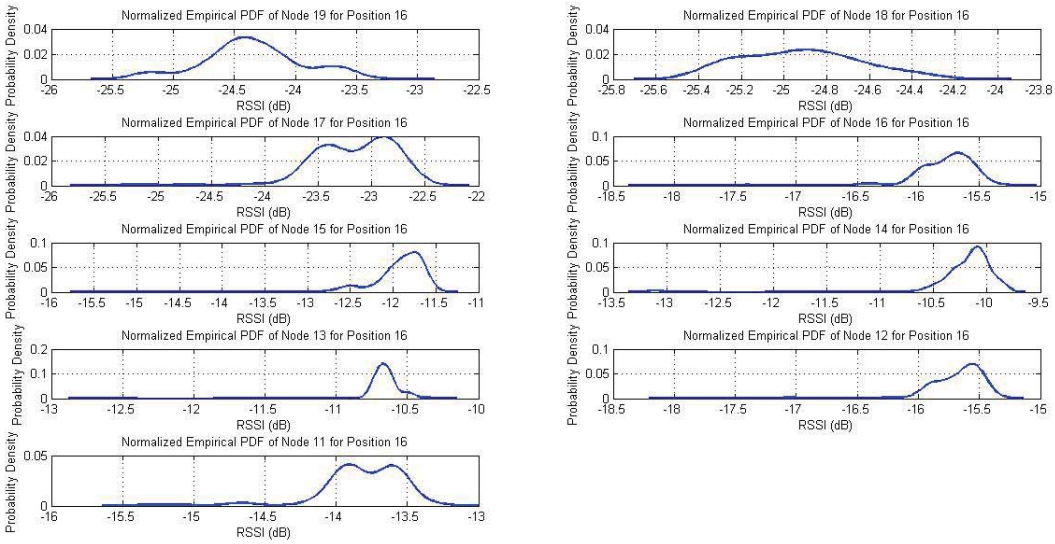

Figure A.16: Histogram of Position 16

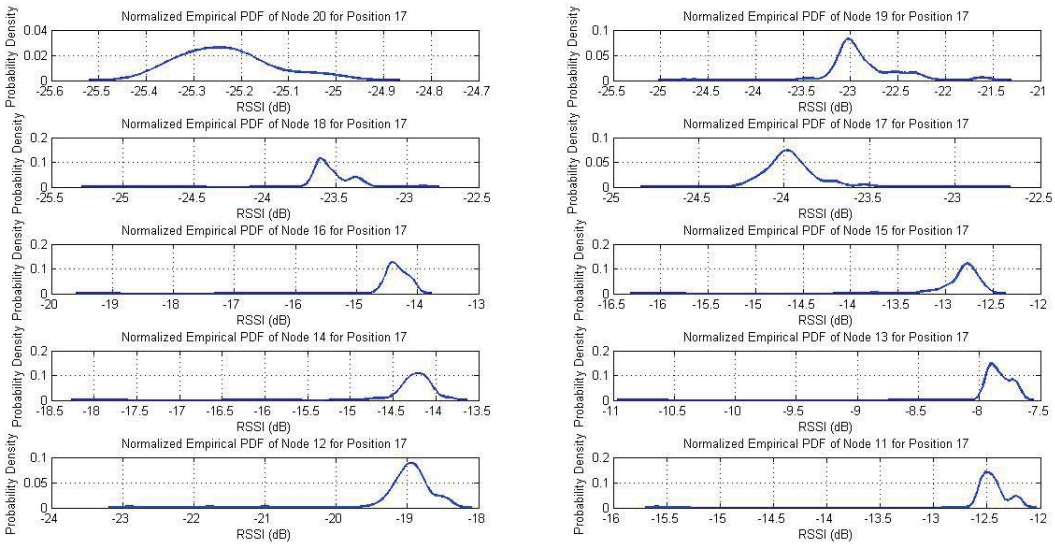

Figure A.17: Histogram of Position 17

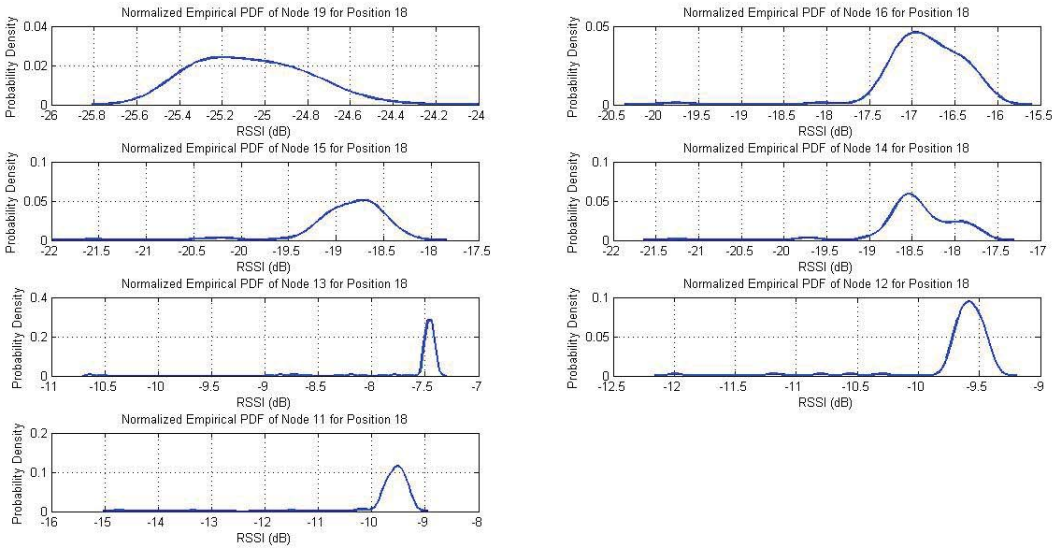

Figure A.18: Histogram of Position 18

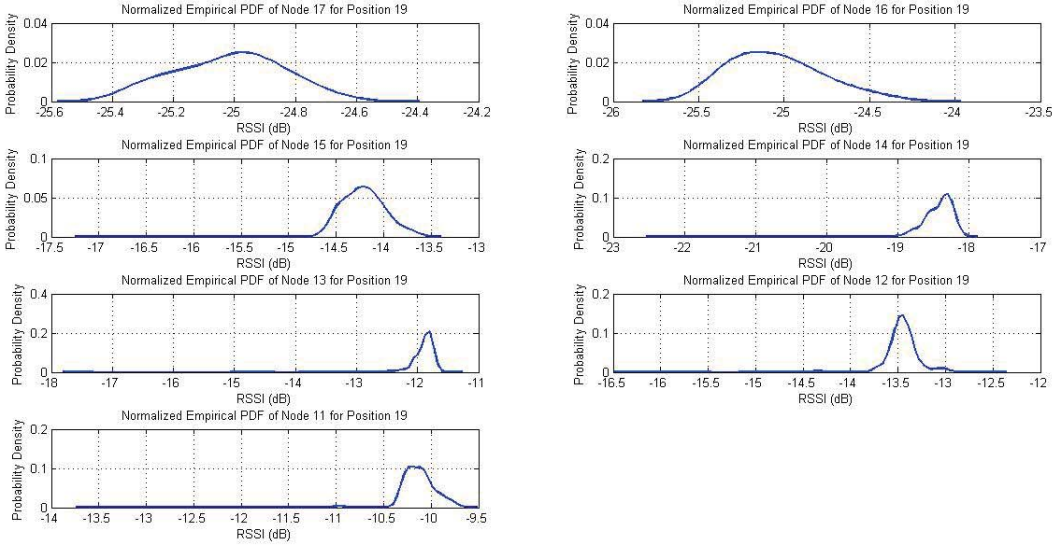

Figure A.19: Histogram of Position 19

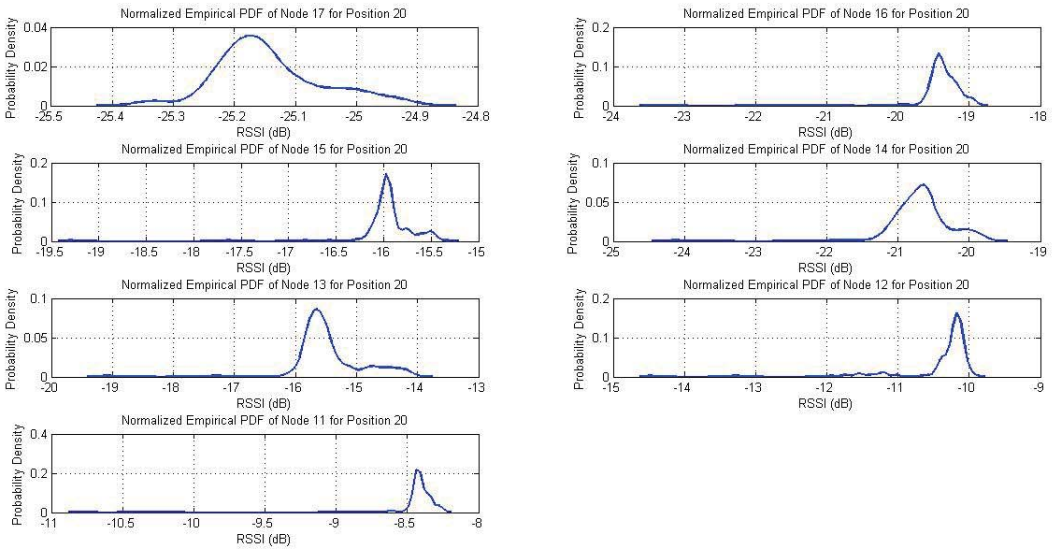

Figure A.20: Histogram of Position 20

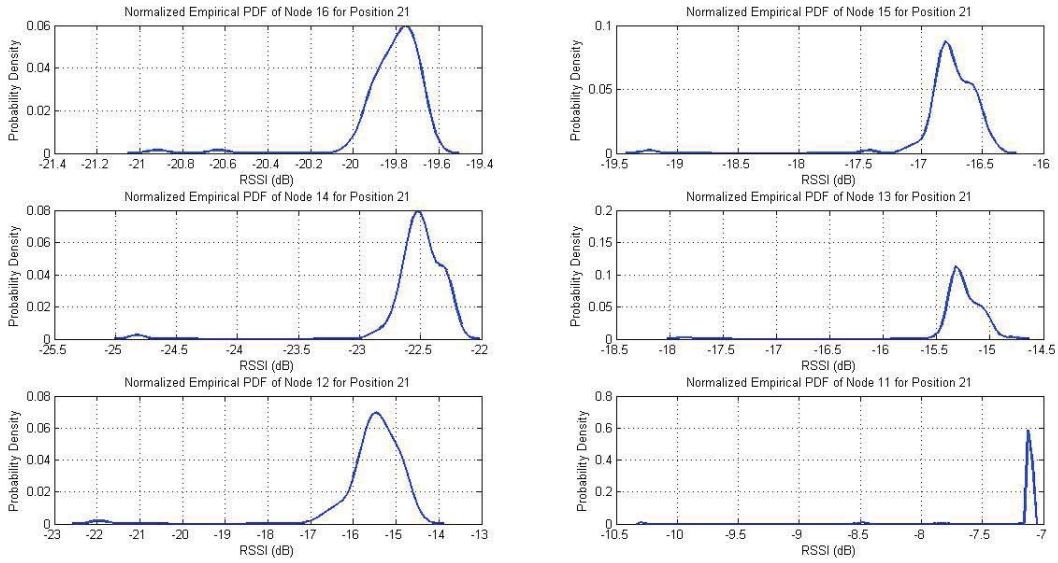

Figure A.21: Histogram of Position 21

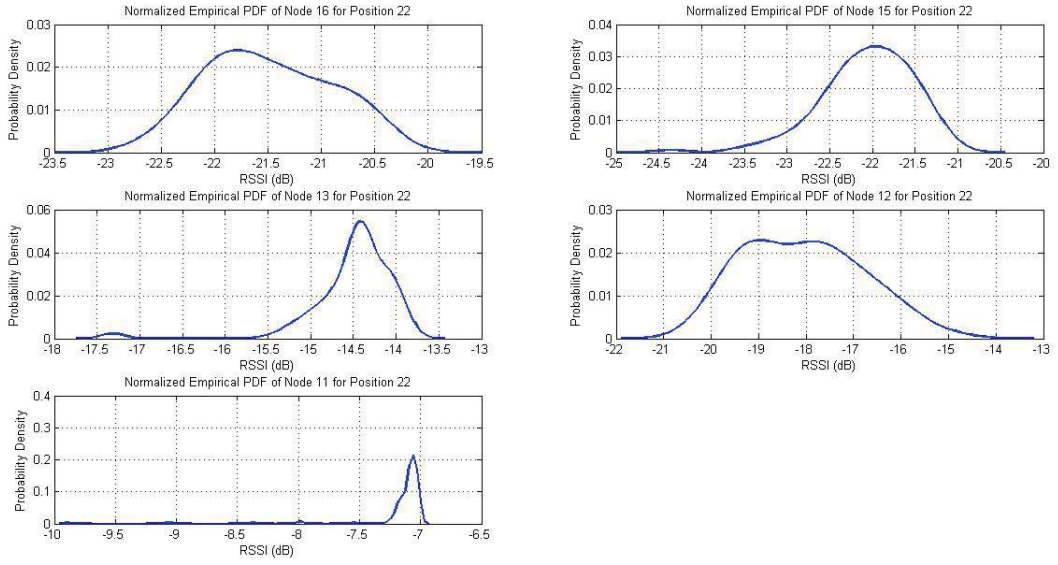

Figure A.22: Histogram of Position 22

Appendix B

Radio Maps using 22 Fingerprints

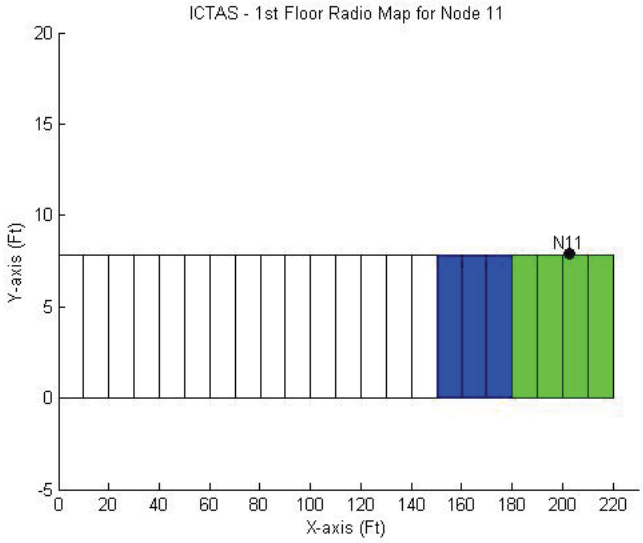

Figure B.1: Radio Map of N11

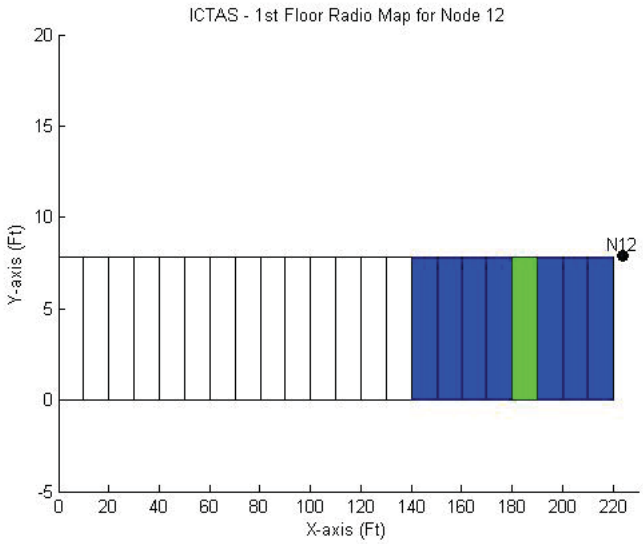

Figure B.2: Radio Map of Node N12

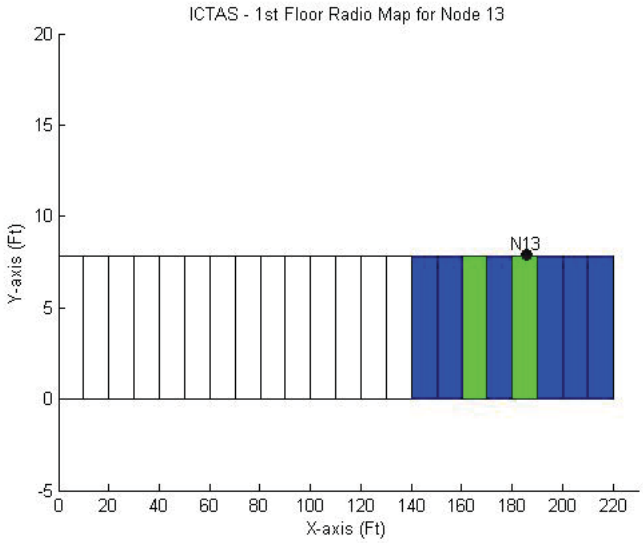

Figure B.3: Radio Map of Node N13

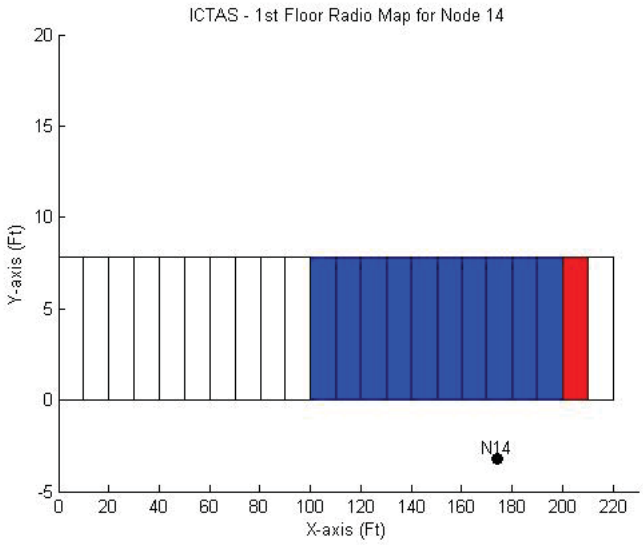

Figure B.4: Radio Map of Node N14

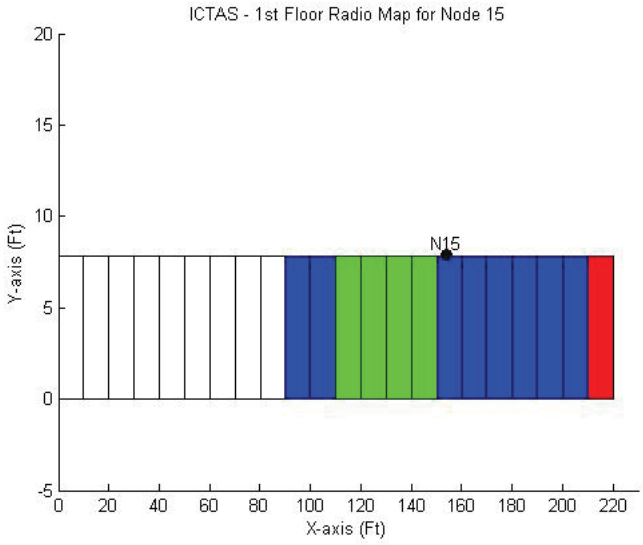

Figure B.5: Radio Map of Node N15

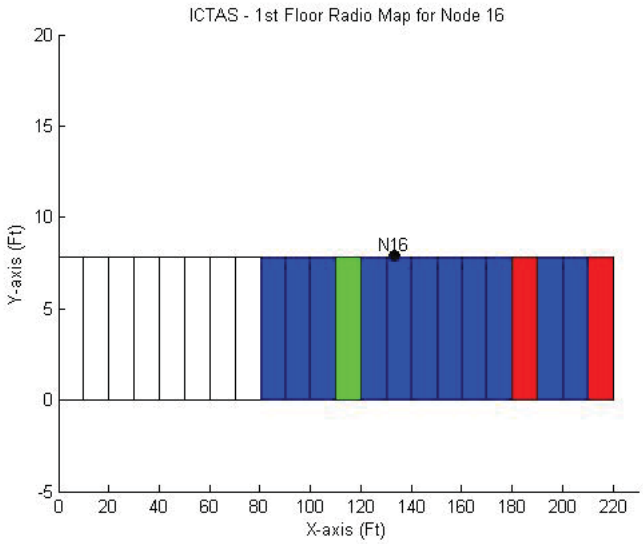

Figure B.6: Radio Map of Node N16

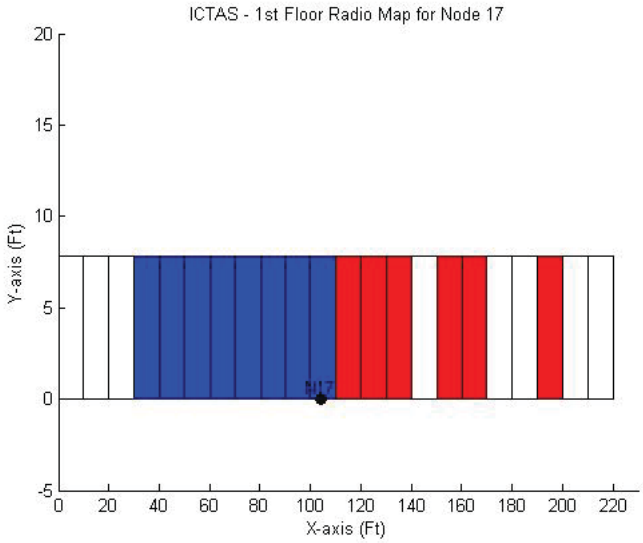

Figure B.7: Radio Map of Node N17

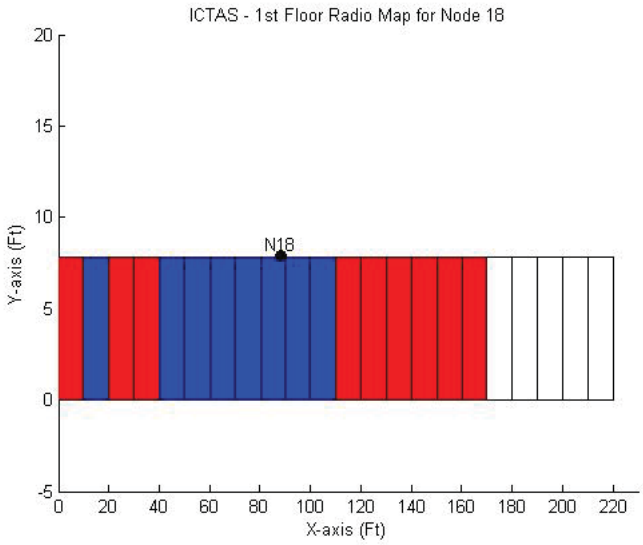

Figure B.8: Radio Map of Node N18

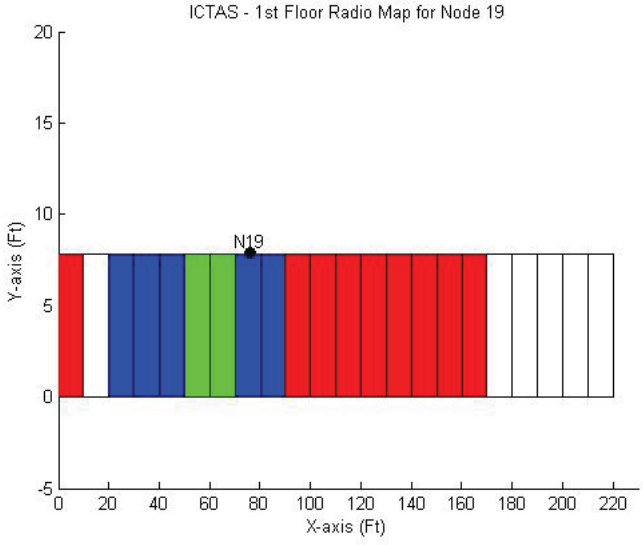

Figure B.9: Radio Map of Node N19

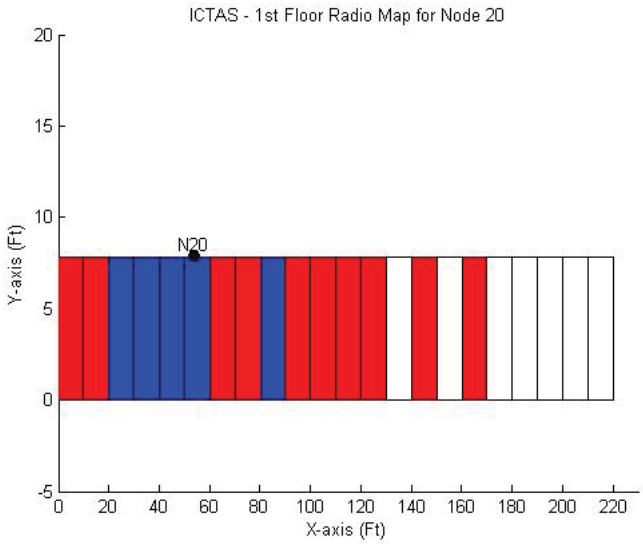

Figure B.10: Radio Map of Node N20

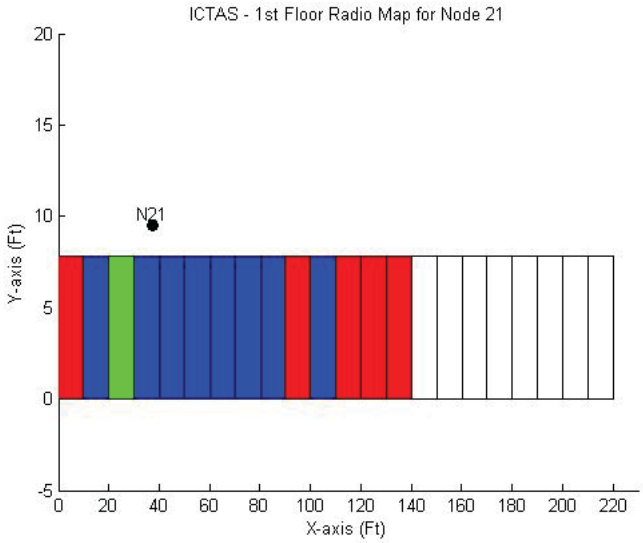

Figure B.11: Radio Map of Node N21

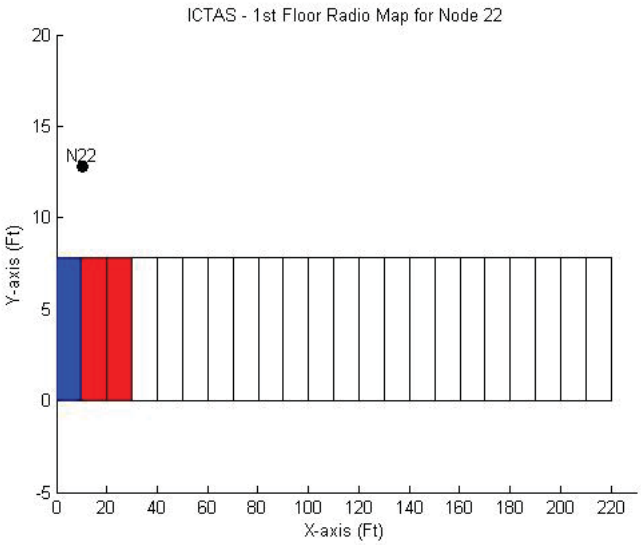

Figure B.12: Radio Map of Node N22

Appendix C

Histograms of a 45 Fingerprint Database

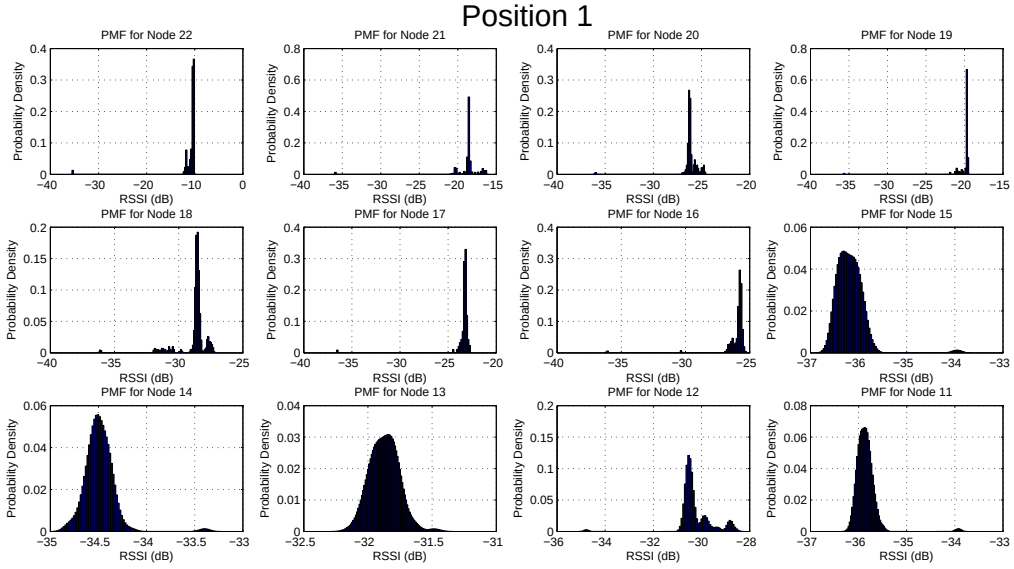

Figure C.1: Histogram of Position 1

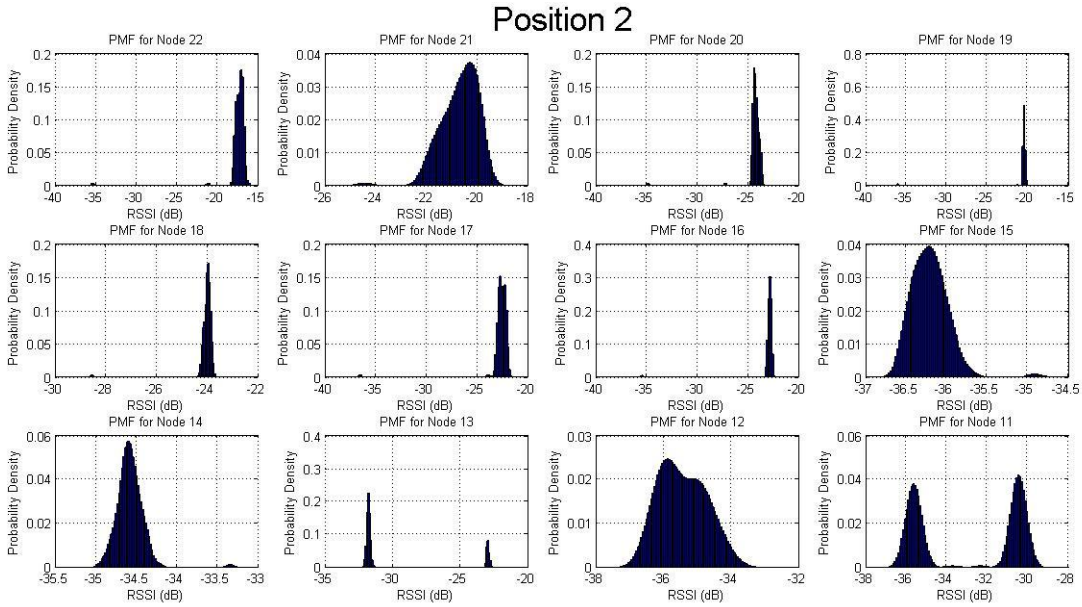

Figure C.2: Histogram of Position 2

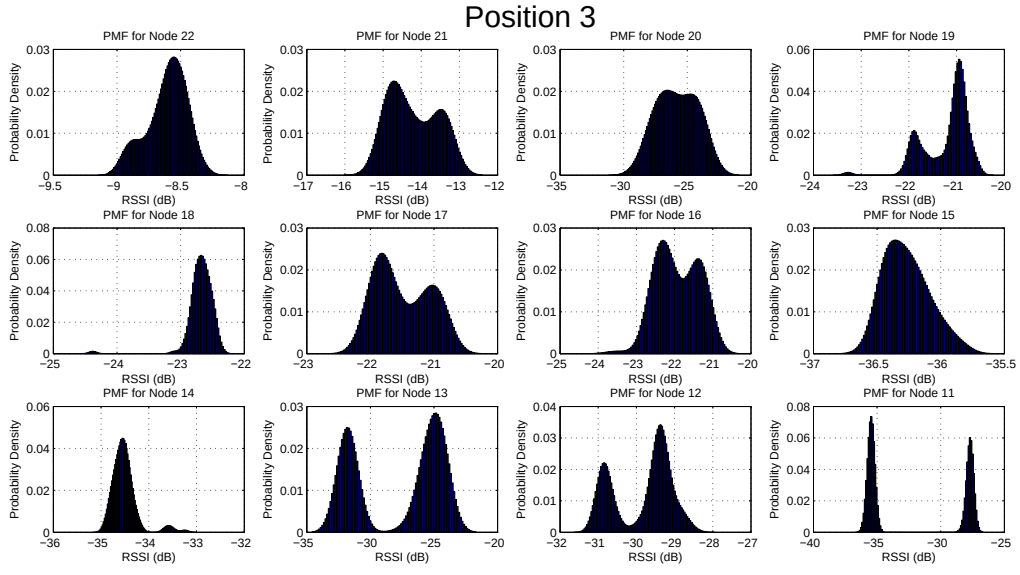

Figure C.3: Histogram of Position 3

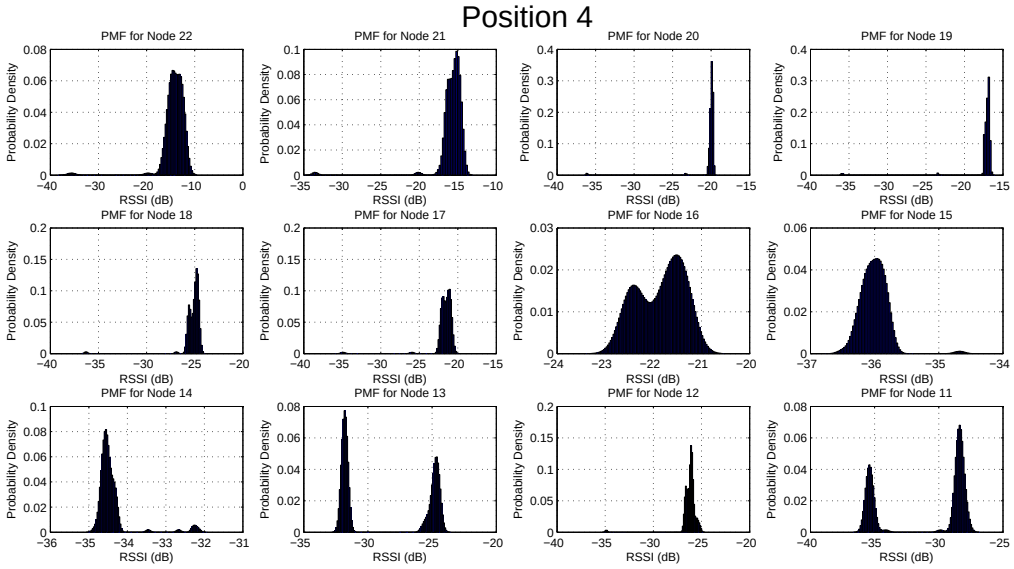

Figure C.4: Histogram of Position 4

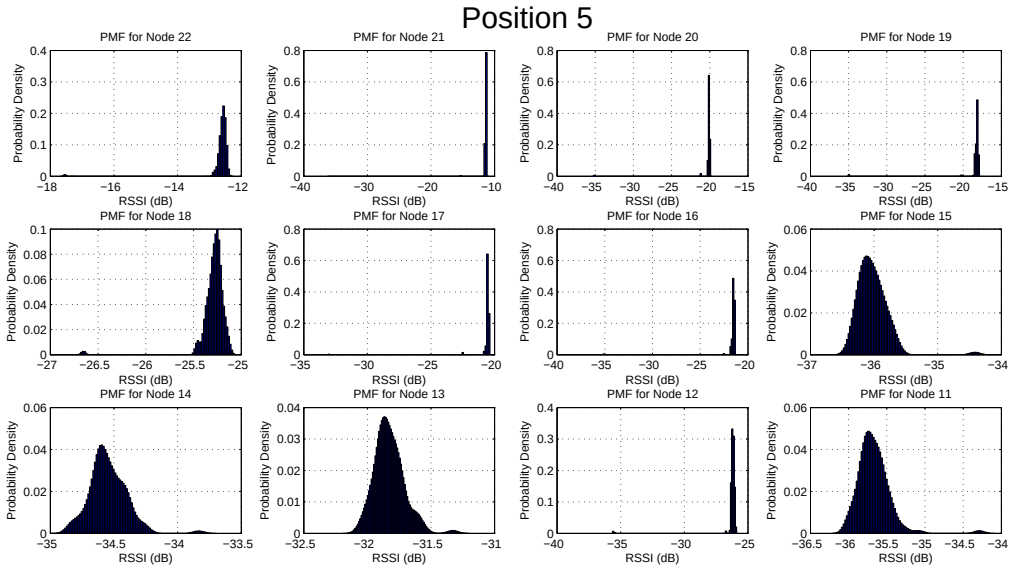

Figure C.5: Histogram of Position 5

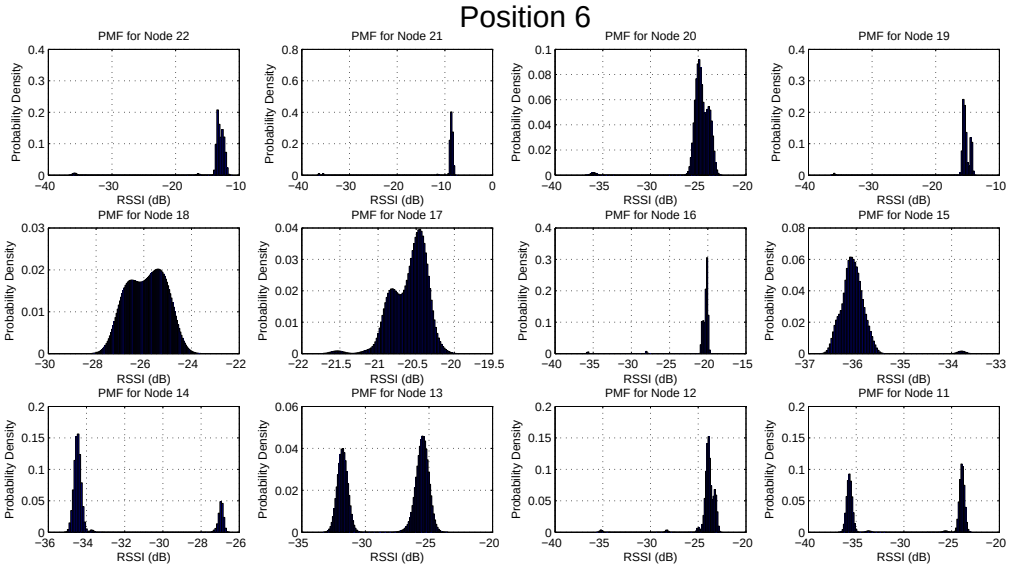

Figure C.6: Histogram of Position 6

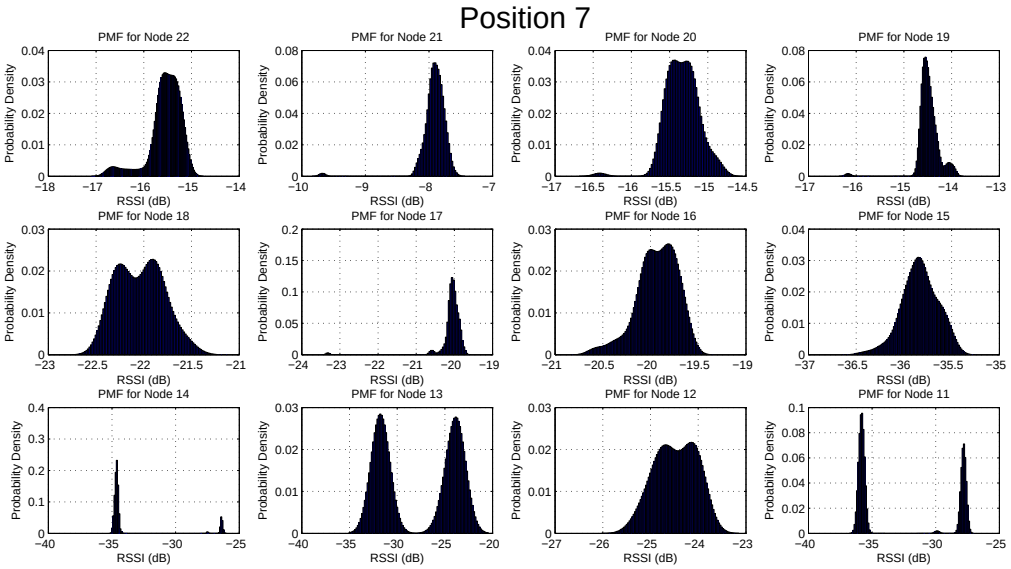

Figure C.7: Histogram of Position 7

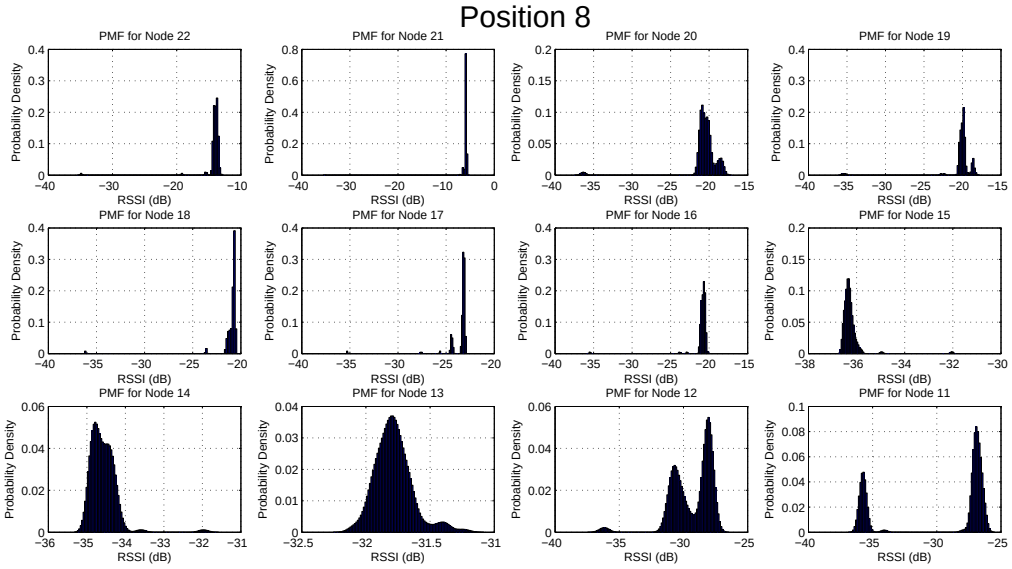

Figure C.8: Histogram of Position 8

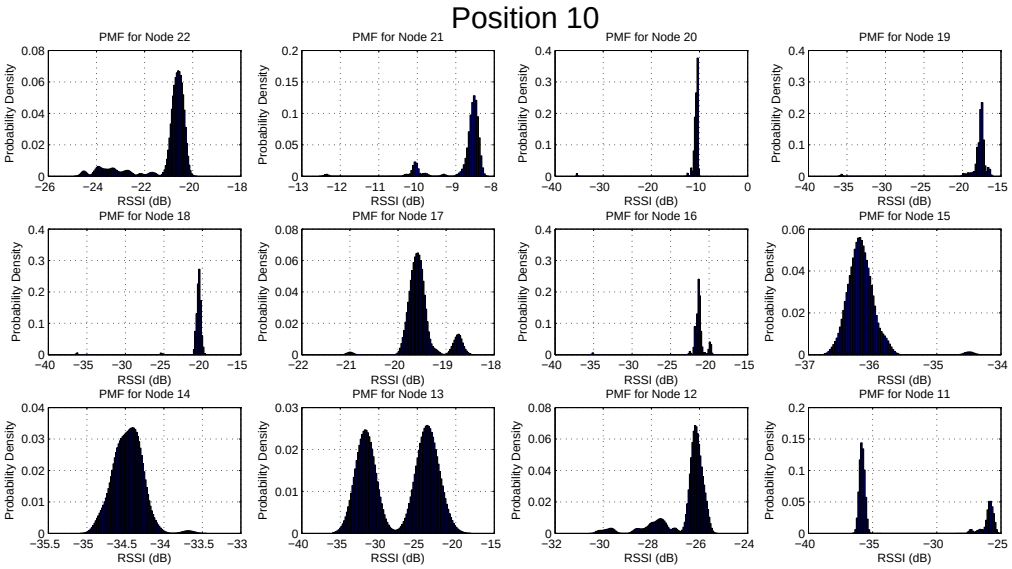

Figure C.9: Histogram of Position 9

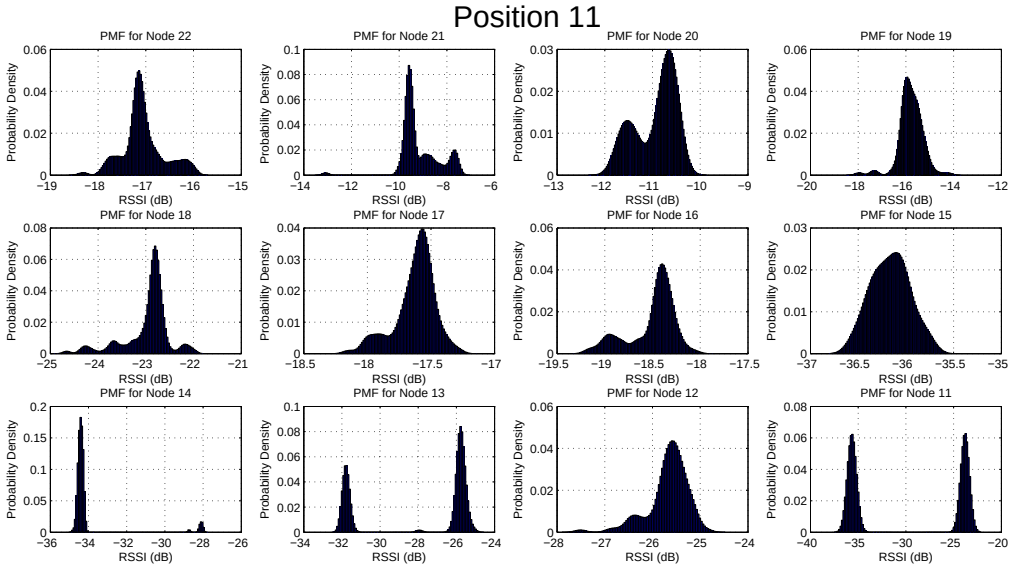

Figure C.10: Histogram of Position 11

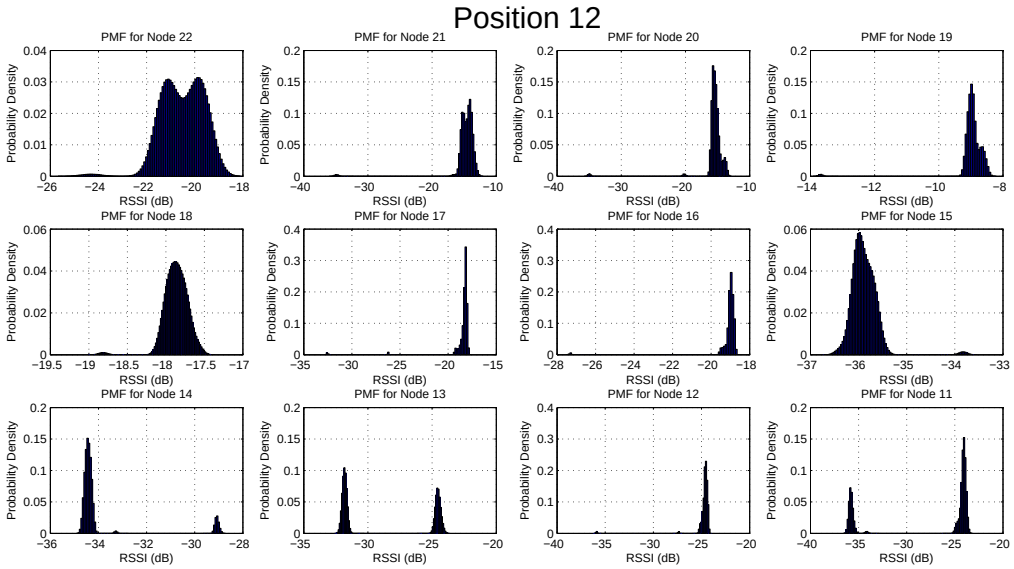

Figure C.11: Histogram of Position 12

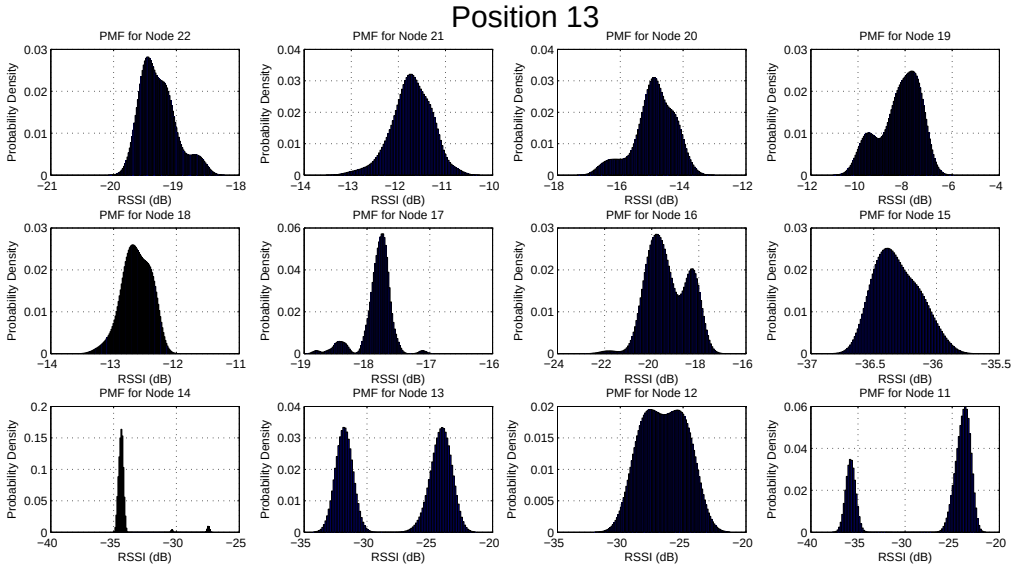

Figure C.12: Histogram of Position 13

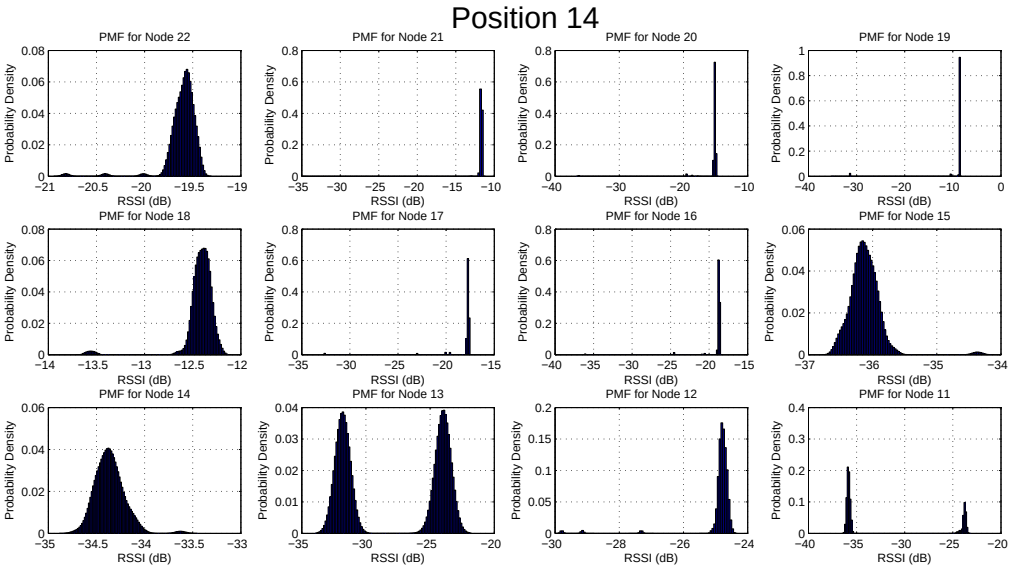

Figure C.13: Histogram of Position 14

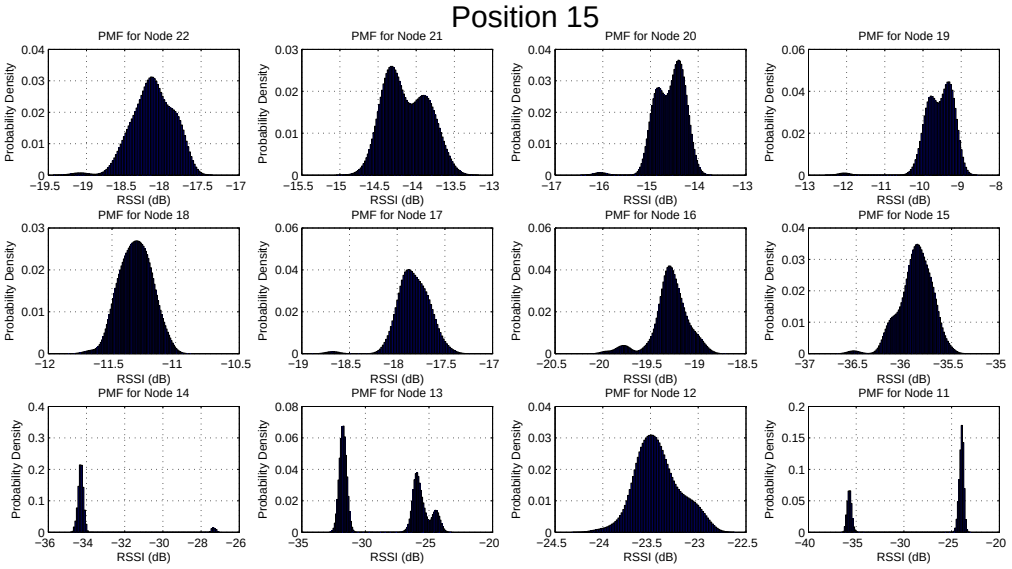

Figure C.14: Histogram of Position 15

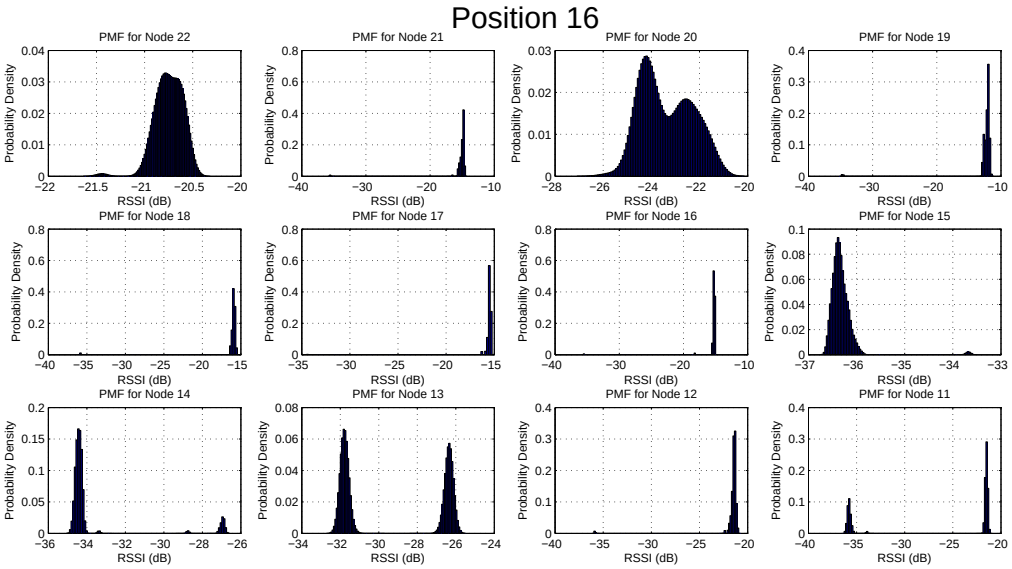

Figure C.15: Histogram of Position 16

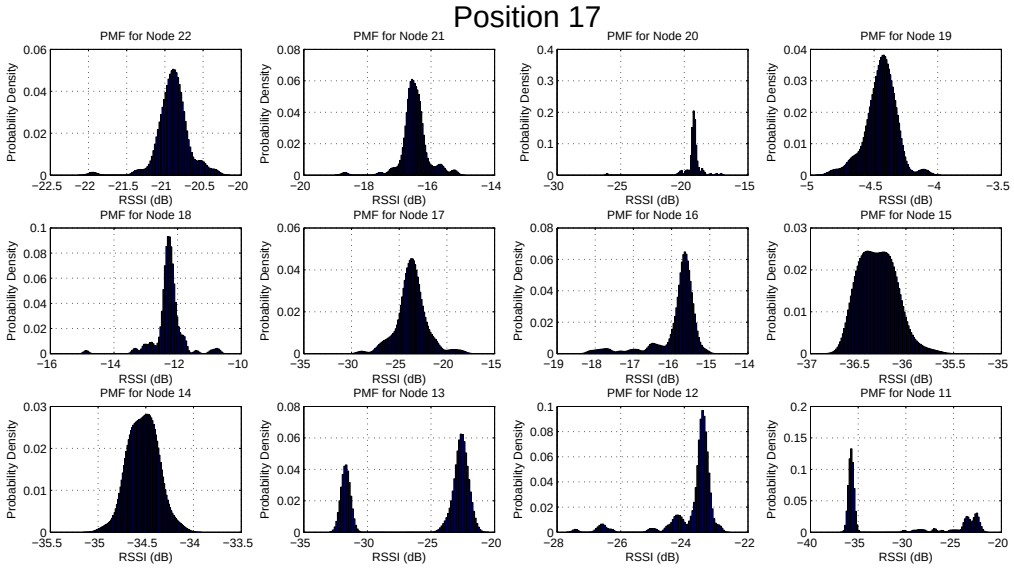

Figure C.16: Histogram of Position 17

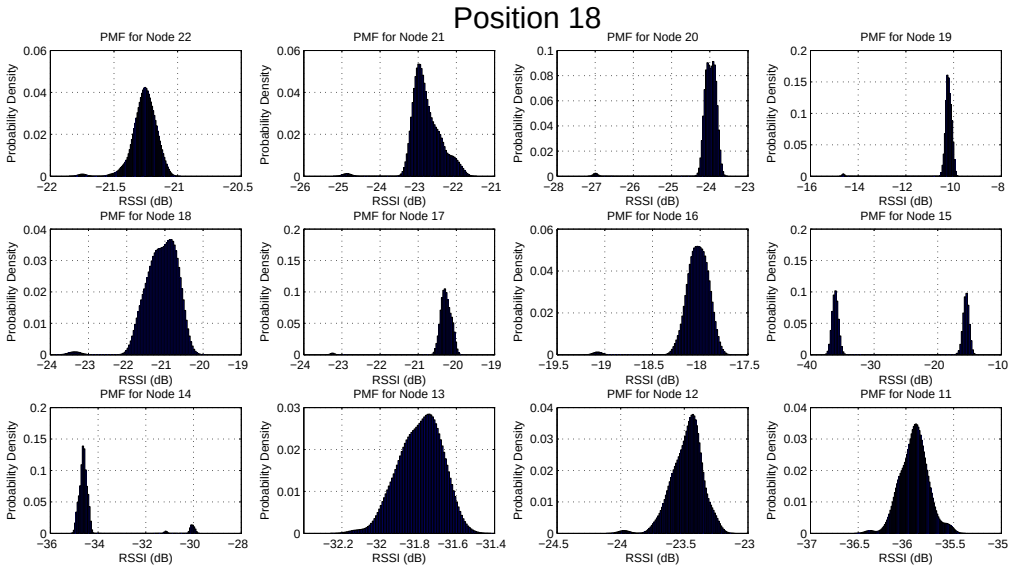

Figure C.17: Histogram of Position 18

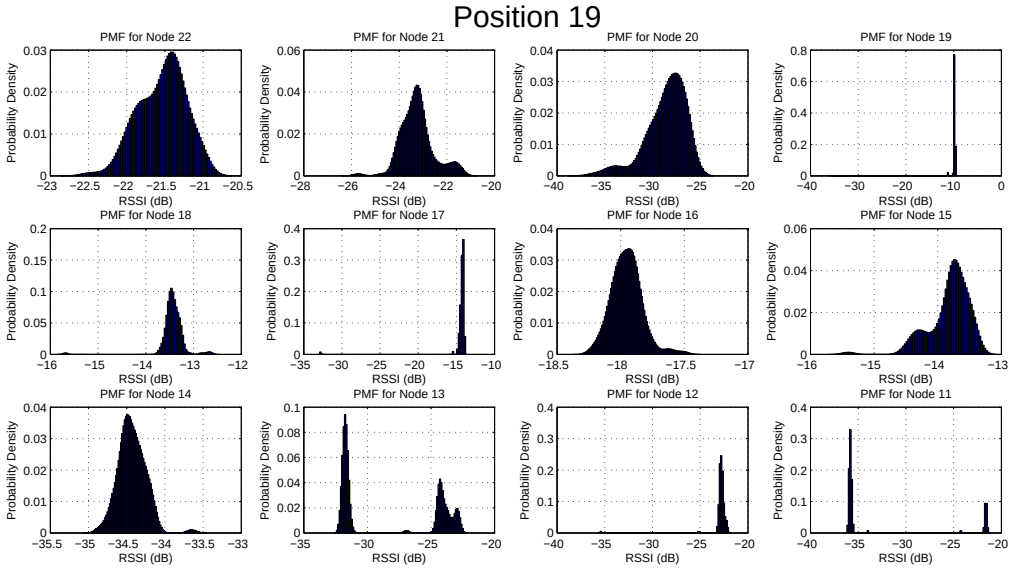

Figure C.18: Histogram of Position 19

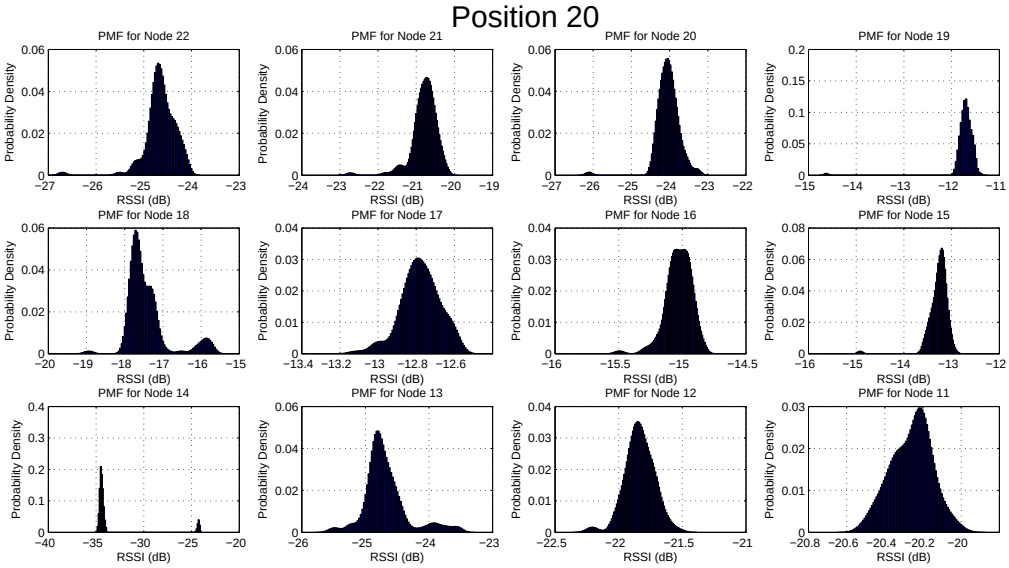

Figure C.19: Histogram of Position 20

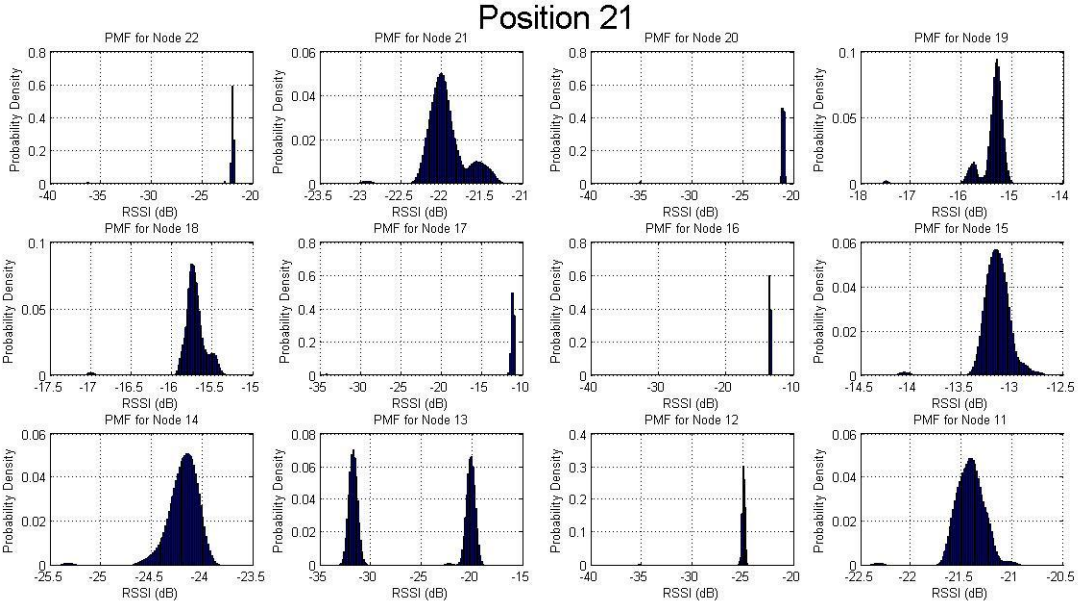

Figure C.20: Histogram of Position 21

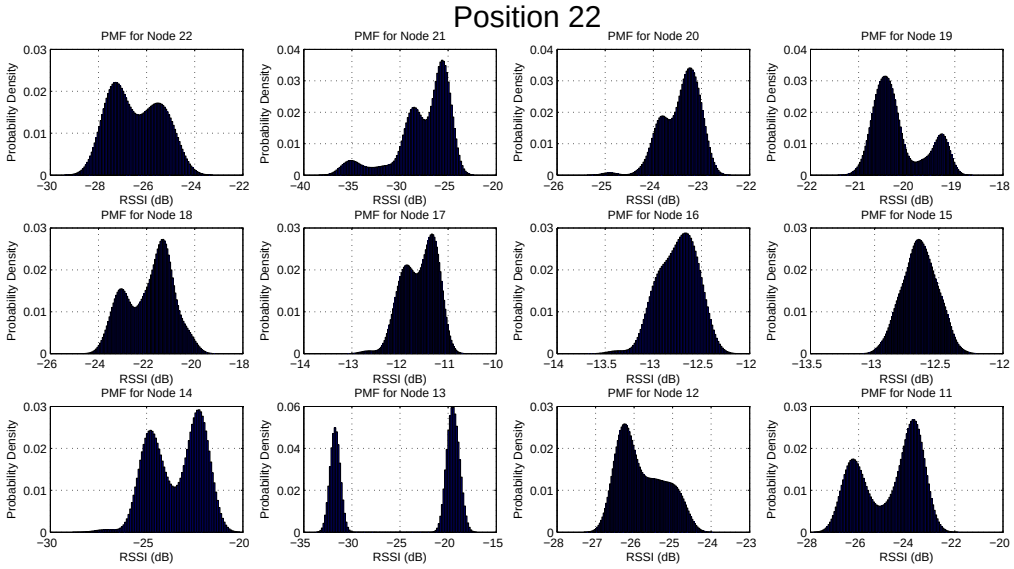

Figure C.21: Histogram of Position 22

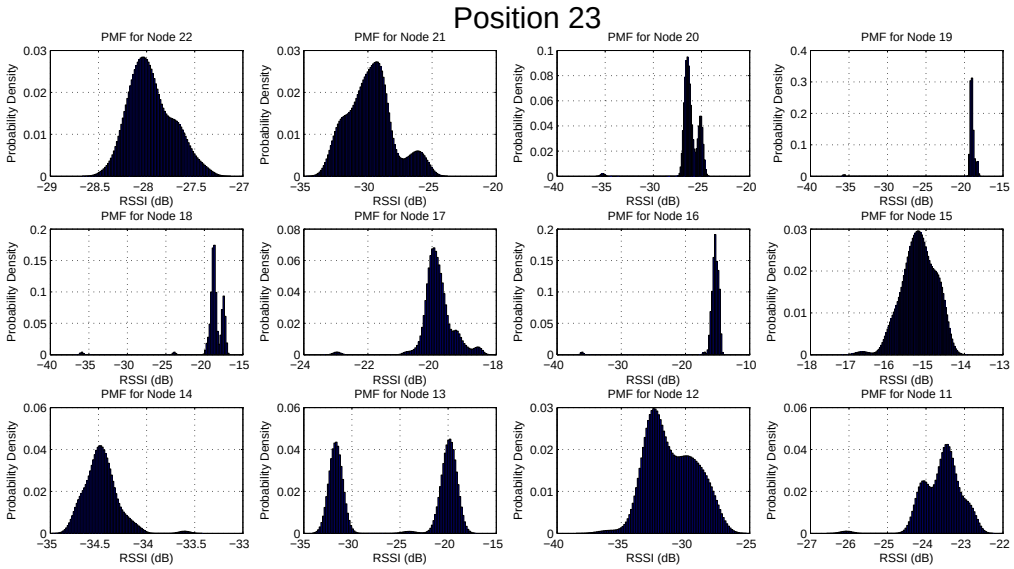

Figure C.22: Histogram of Position 23

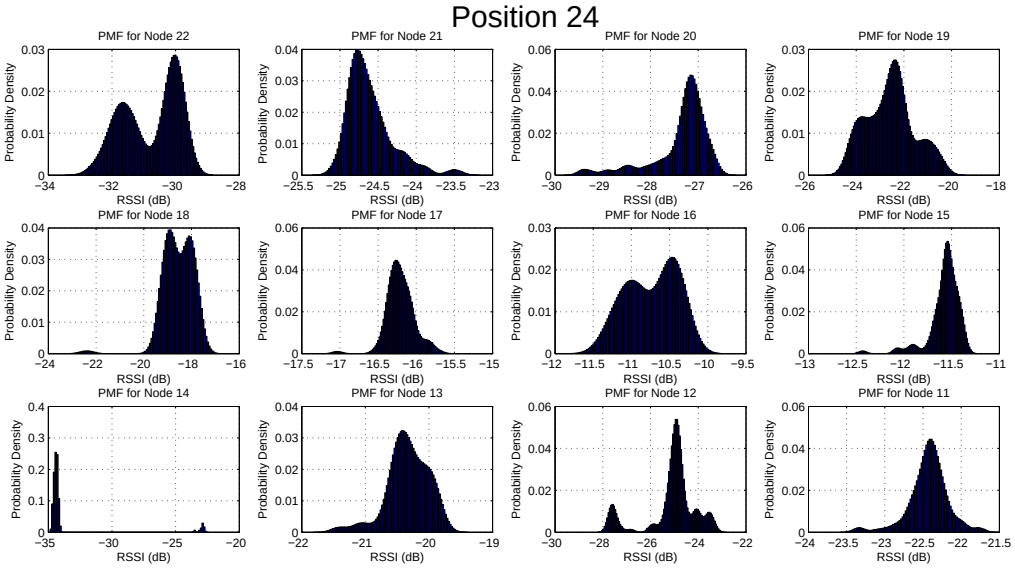

Figure C.23: Histogram of Position 24

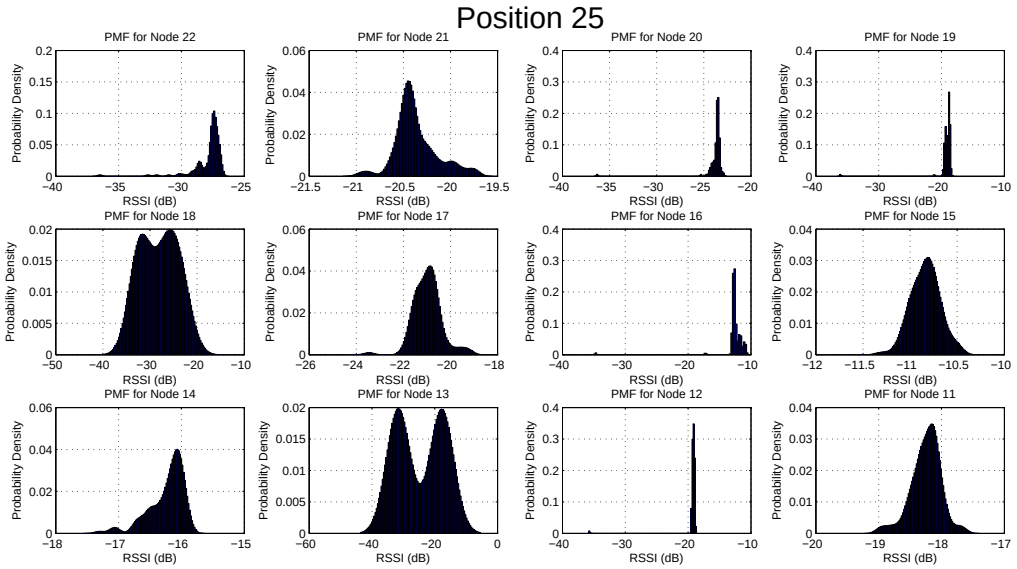

Figure C.24: Histogram of Position 25

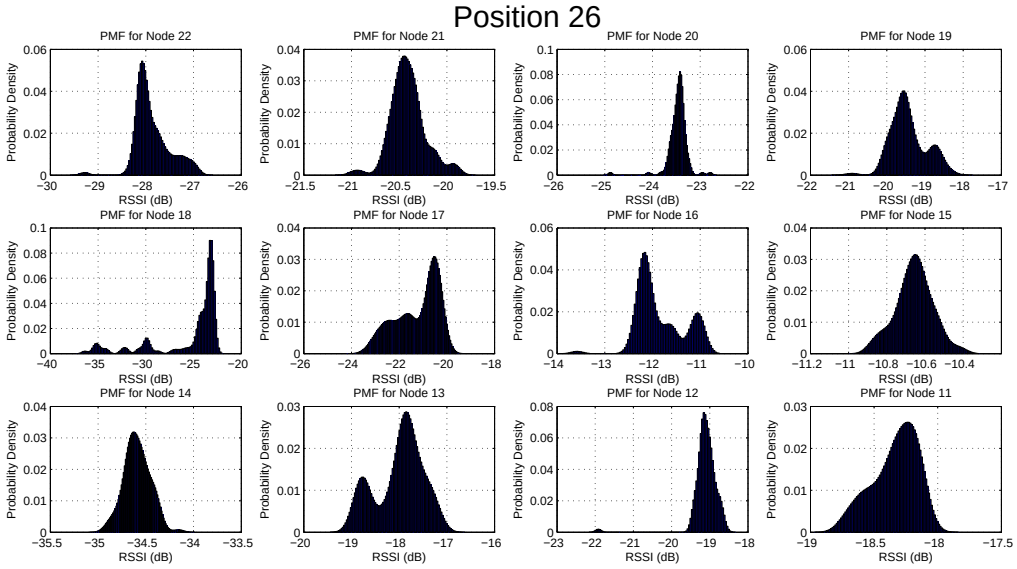

Figure C.25: Histogram of Position 26

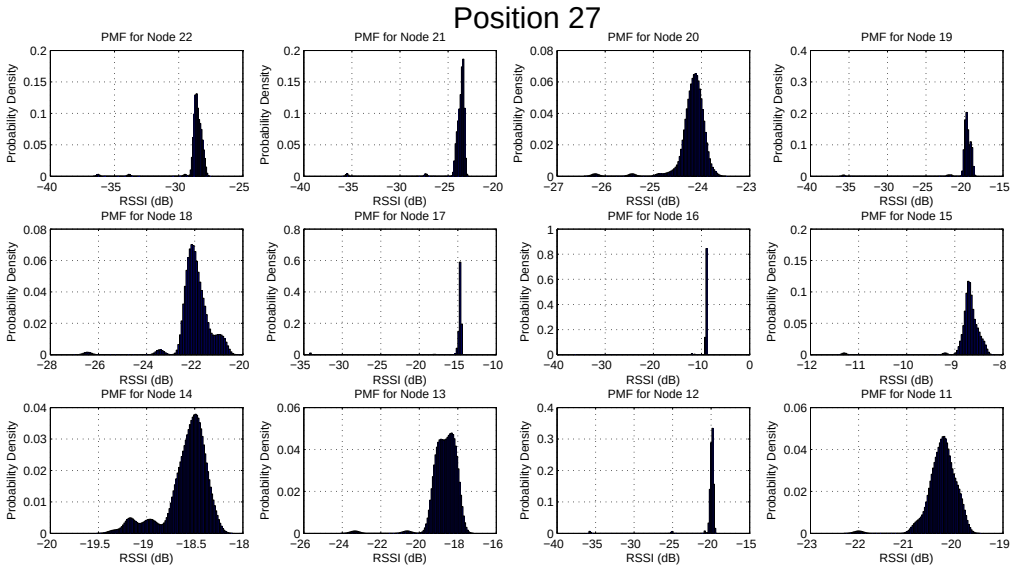

Figure C.26: Histogram of Position 27

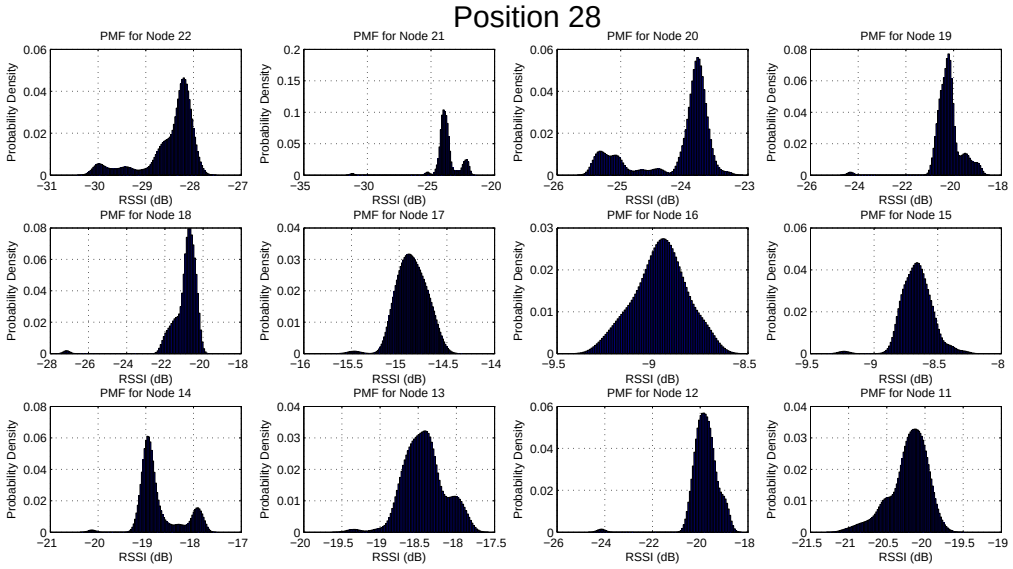

Figure C.27: Histogram of Position 28

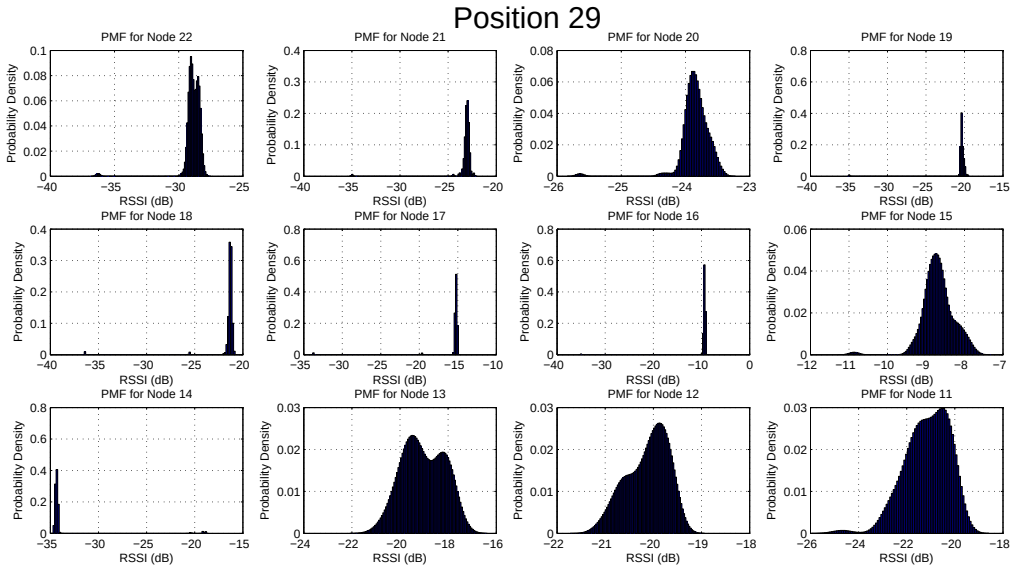

Figure C.28: Histogram of Position 29

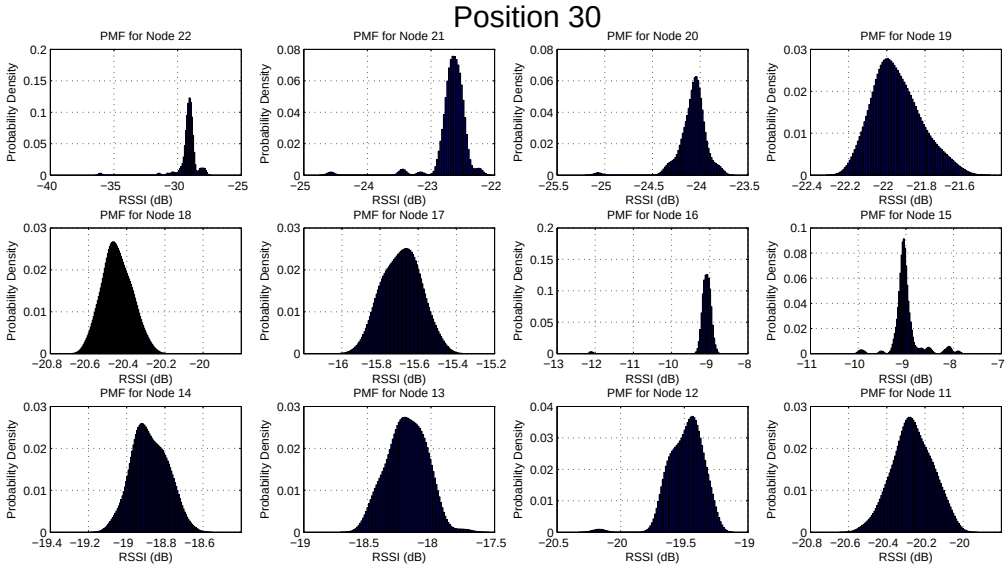

Figure C.29: Histogram of Position 30

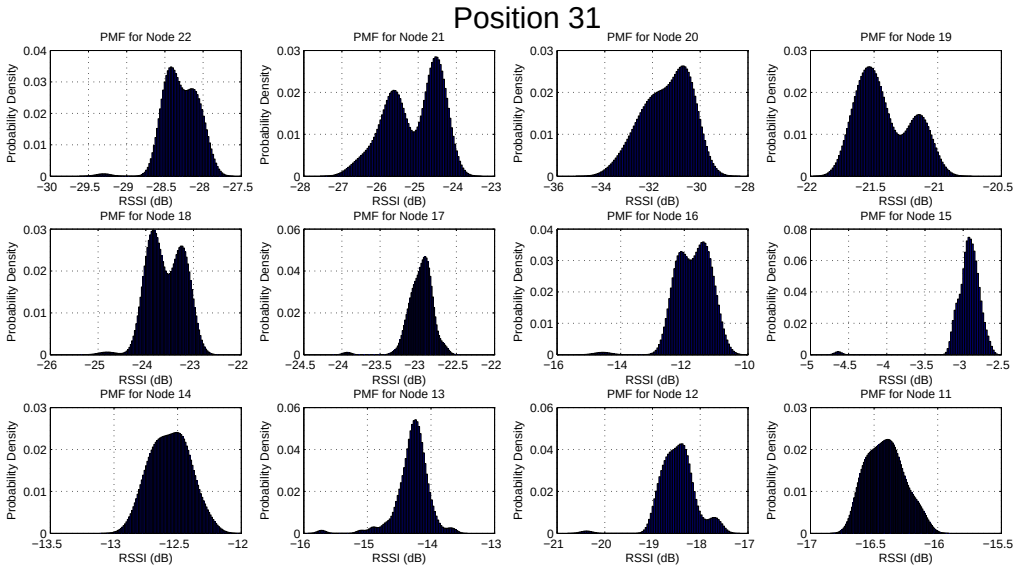

Figure C.30: Histogram of Position 31

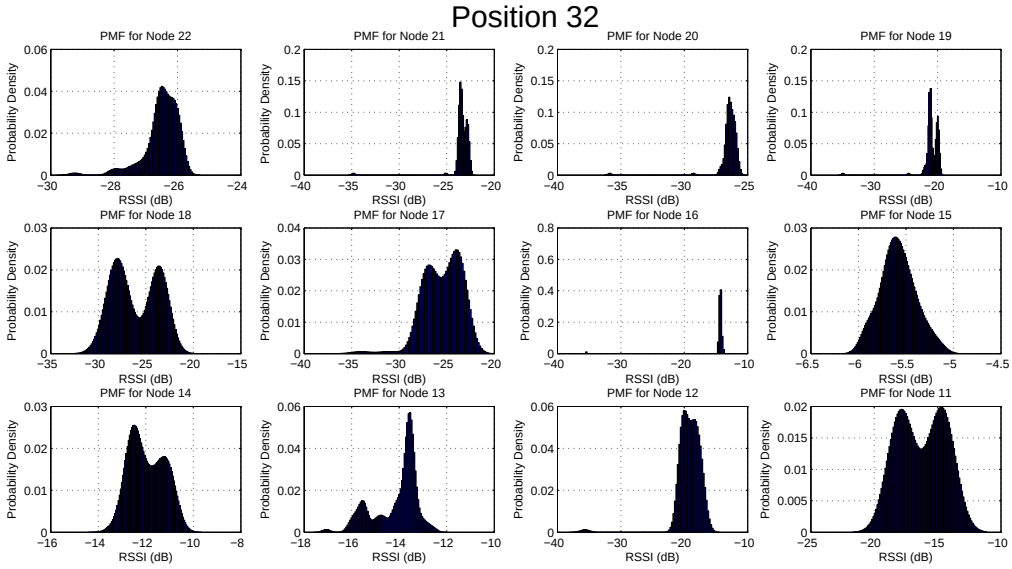

Figure C.31: Histogram of Position 32

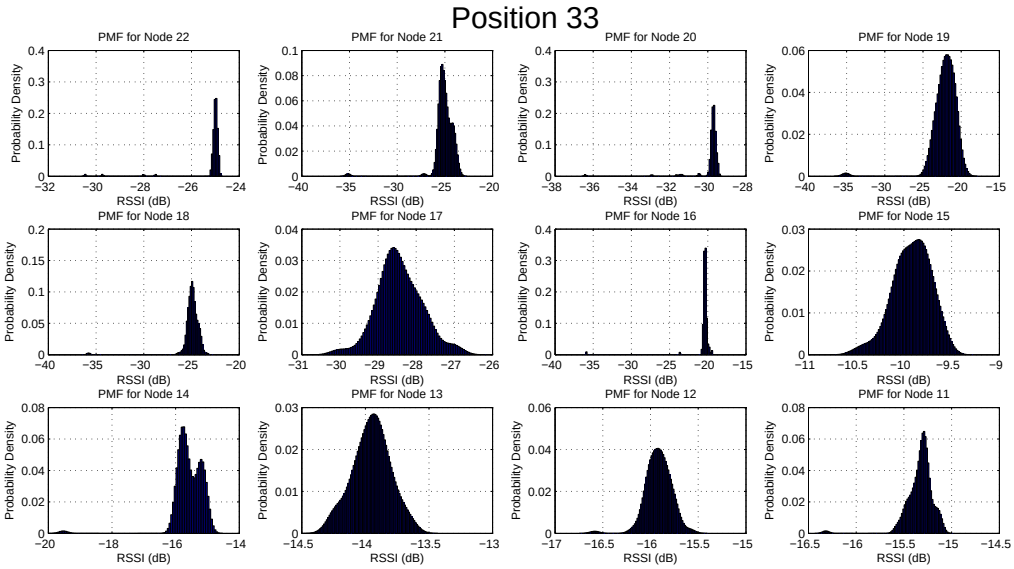

Figure C.32: Histogram of Position 33

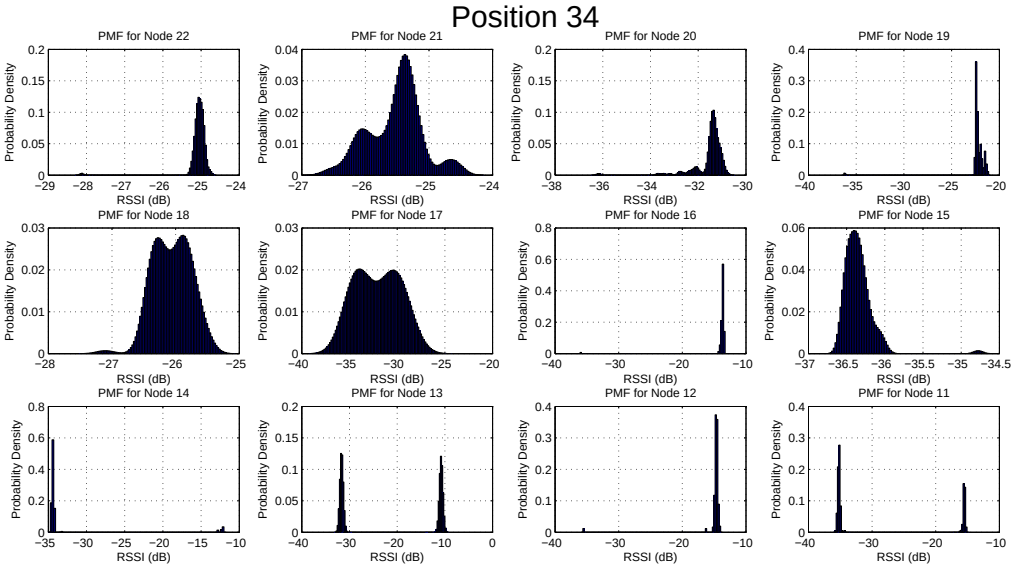

Figure C.33: Histogram of Position 34

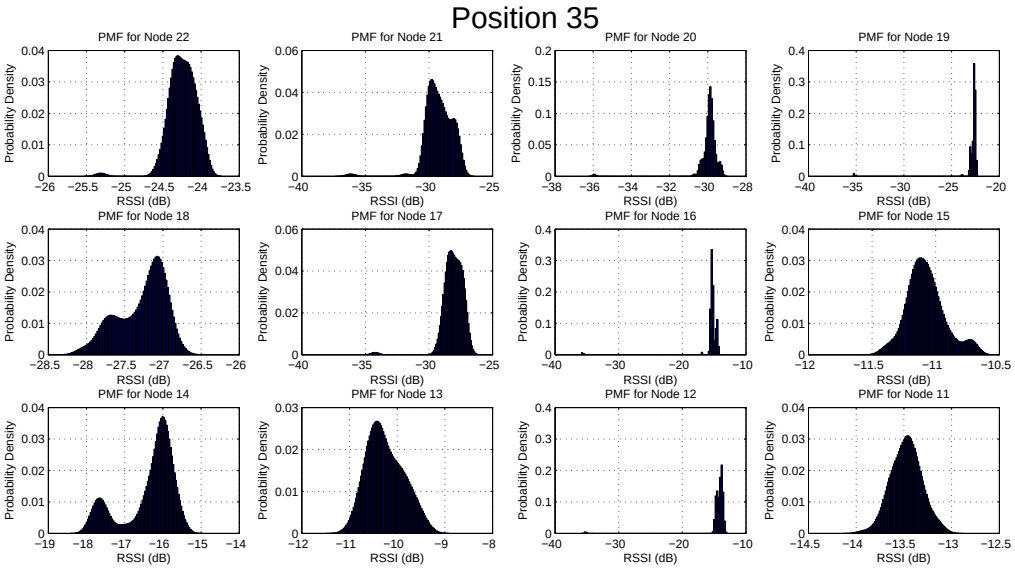

Figure C.34: Histogram of Position 35

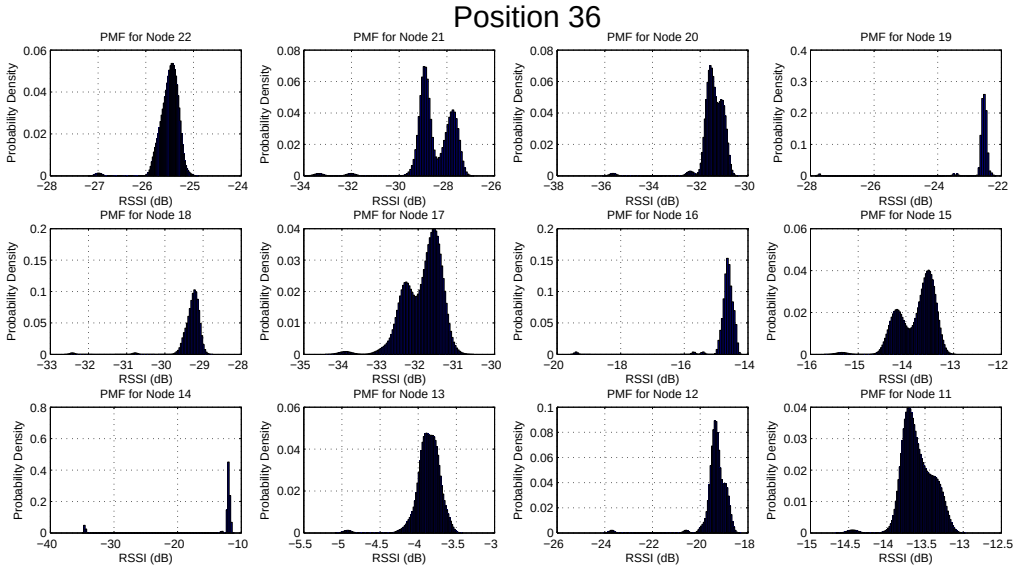

Figure C.35: Histogram of Position 36

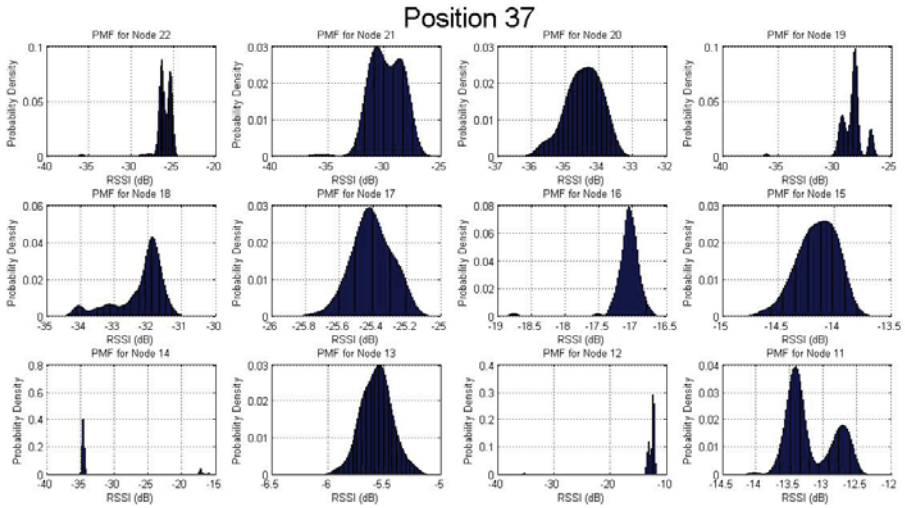

Figure C.36: Histogram of Position 37

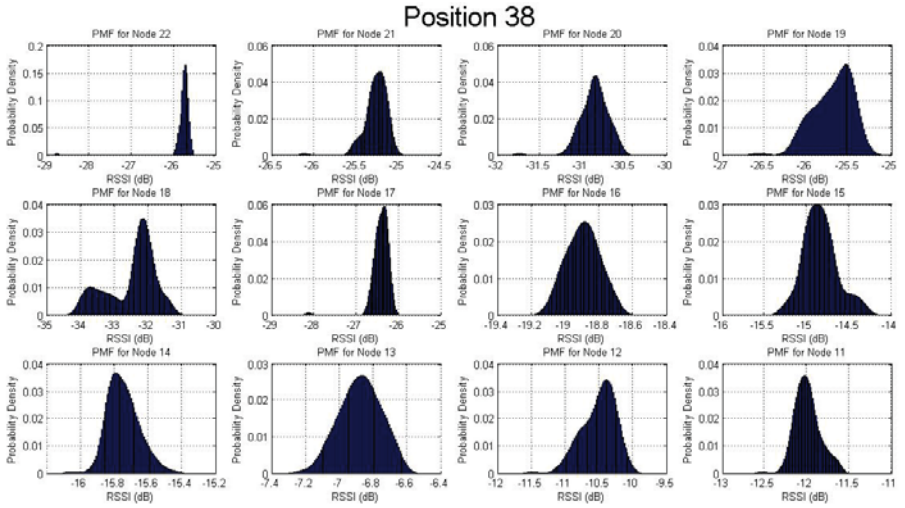

Figure C.37: Histogram of Position 38

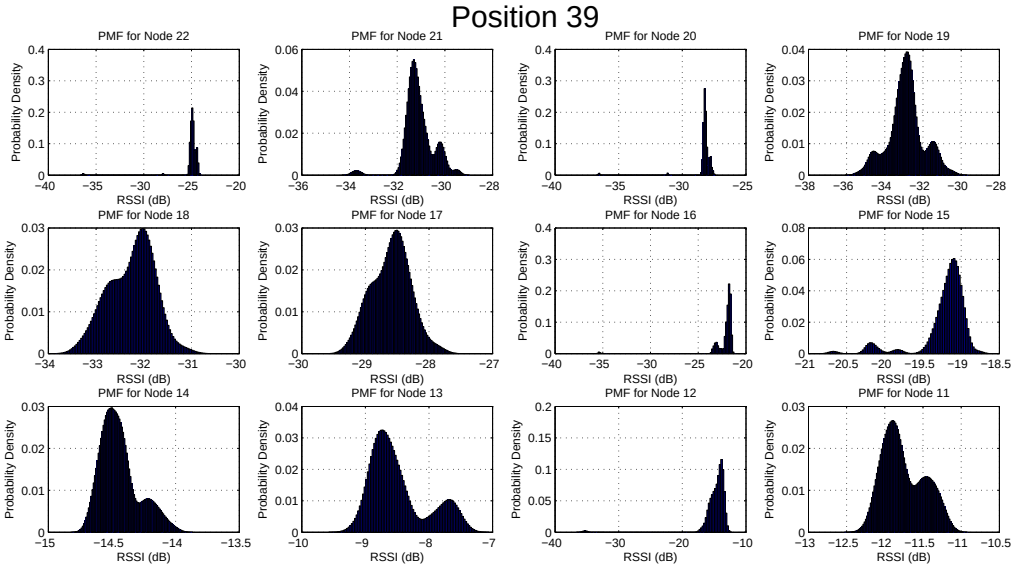

Figure C.38: Histogram of Position 39

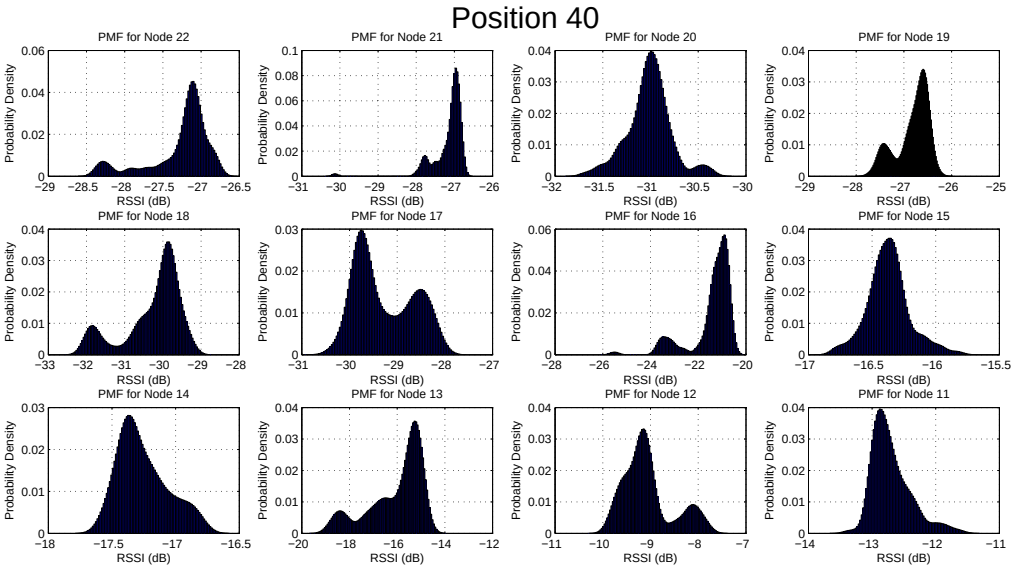

Figure C.39: Histogram of Position 40

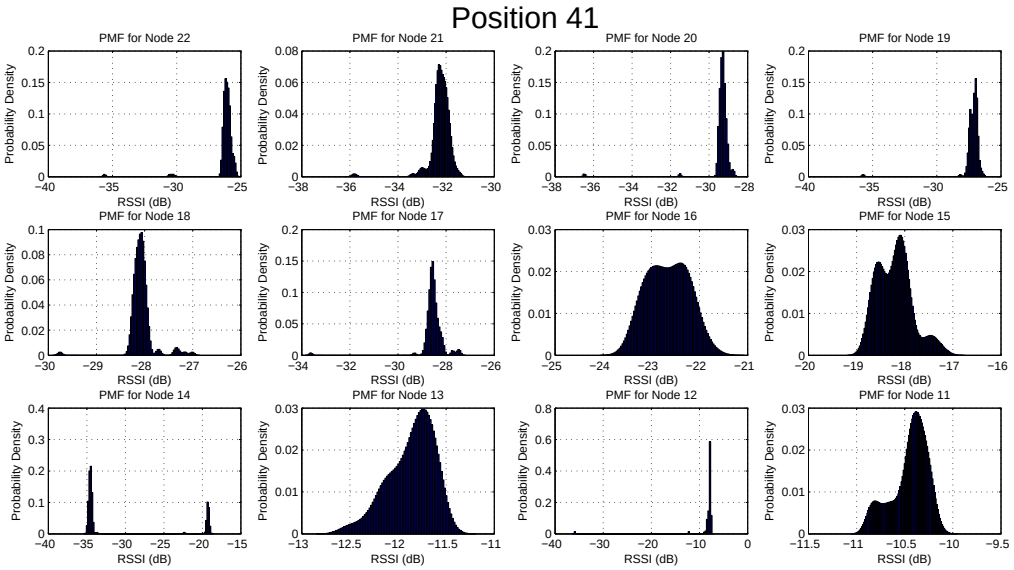

Figure C.40: Histogram of Position 41

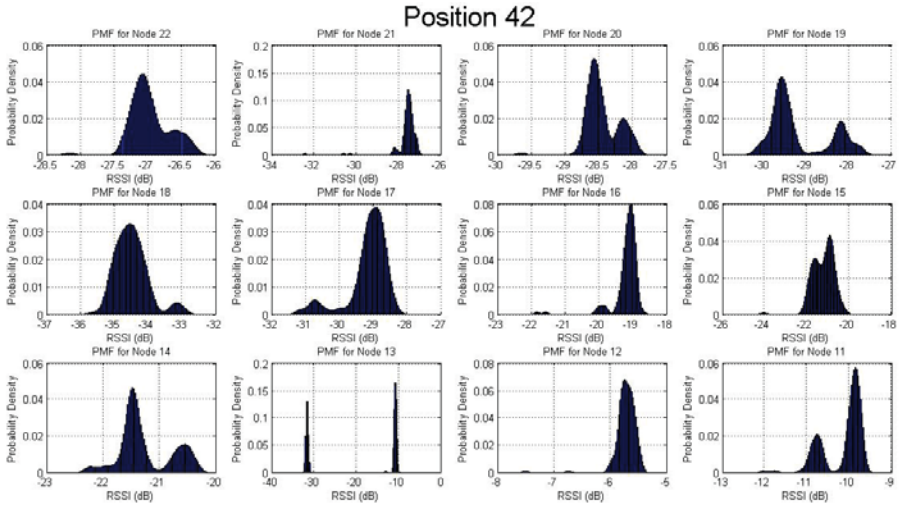

Figure C.41: Histogram of Position 42

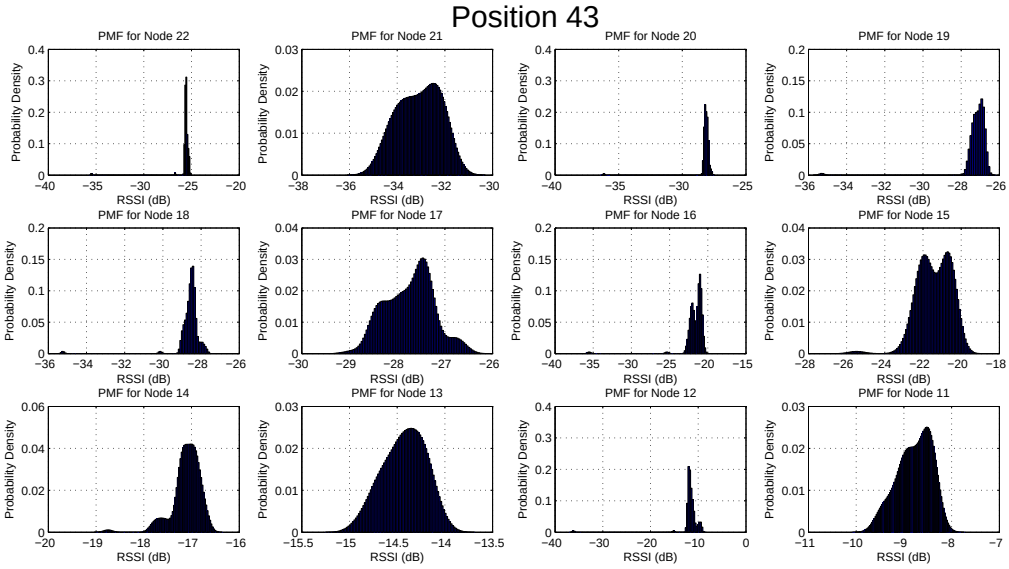

Figure C.42: Histogram of Position 43

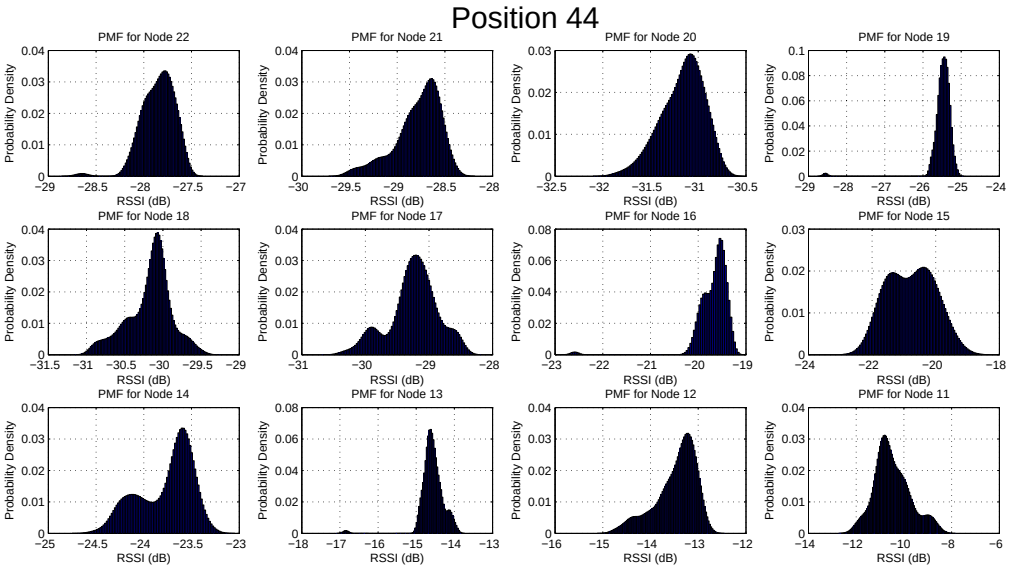

Figure C.43: Histogram of Position 44

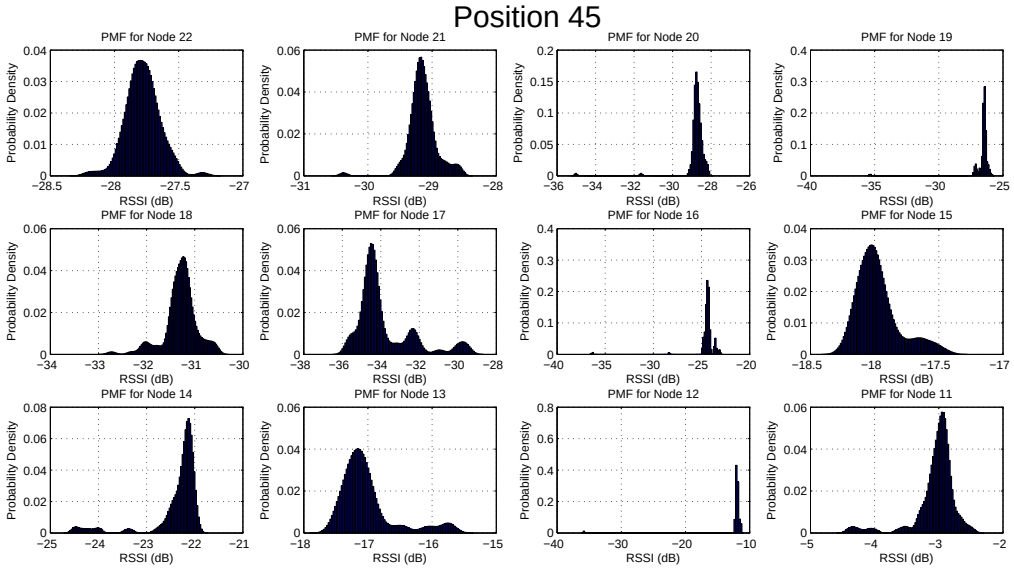

Figure C.44: Histogram of Position 45

Appendix D

Table of Correlation Coefficients

Table D.1: Correlation Coefficient for N22 in Scenario 1

Position	N22, N21	N22, N20	N22, N19	N22, N18
1	0.0849	0.5871	0.986	0.392
2	0.6244	-0.2268	N/A	0.2879
3	-0.2348	0.2521	-0.2269	-0.2784

Table D.2: Correlation Coefficient for N21 in Scenario 1

Position	N21, N20	N21, N19	N21, N18	N21, N17
1	0.0475	0.2024	-0.6357	N/A
2	-0.5457	N/A	0.7736	N/A
3	-0.5397	0.9997	0.9975	N/A
4	0.4309	0.2571	-0.5565	0.2813
5	-0.3204	-0.627	-0.4605	0.8626
6	0.0478	0.0887	-0.6304	0.8244
7	-0.2447	0.9889	0.6387	-0.3023
8	0.3403	-0.3308	-0.819	-0.5985
9	-0.3369	-0.5644	-0.0167	-0.5198
10	-	0.0324	-0.1691	-0.195
11	-0.6422	-0.2225	-0.225	-0.635
12	-0.324	0.4197	-0.4926	-0.1097
13	-0.3644	0.1439	-0.0262	0.5489
14	N/A	-0.3426	-0.4098	0.1958

Table D.3: Correlation Coefficient for N22 in Scenario 3

Position	**N22, N21**	**N22, N20**	**N22, N19**	**N22, N18**	**N22, N17**
1	0.8312	0.8166	0.9878	0.5932	0.9861
2	0.6173	0.9677	0.9730	0.9085	0.9822
3	0.3405	0.1772	-0.1497	-0.2136	0.1127
4	0.8372	0.8898	0.8830	0.895	0.9479
5	0.9791	0.9815	0.9827	0.8760	0.9817
40	0.6300	-0.5249	0.5755	0.8798	-0.0814
41	0.0049	0.3280	0.3994	0.2561	0.3632
42	-0.0751	0.8538	0.8193	0.5037	-0.5469
43	0.12729	0.9825	0.9452	0.9404	0.2895
44	0.3861	0.7616	0.3765	0.0307	0.2182
45	0.4346	0.4475	0.3429	0.5023	0.3765

Table D.4: Correlation Coefficient for Scenario 4 at Position 13

	N22
N21	0.0101
N20	0.2455
N19	-0.1008
N18	0.2333
N17	-0.0314
N16	-0.0516
N15	0.2564
N14	0.0457
N13	0.0721
N12	-0.0401
N11	0.0979

Appendix E

Radio Maps using 45 Fingerprints

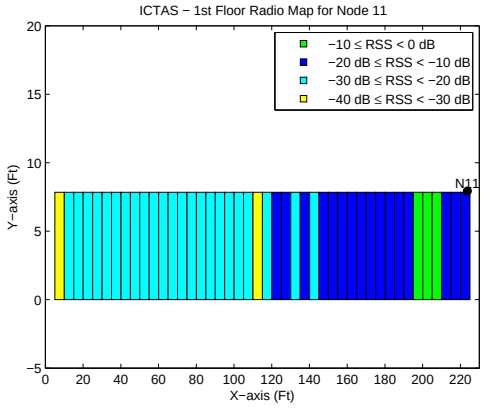

Figure E.1: Radio Map of N11

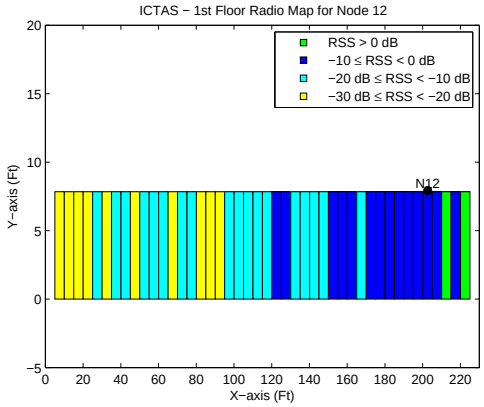

Figure E.2: Radio Map of Node N12

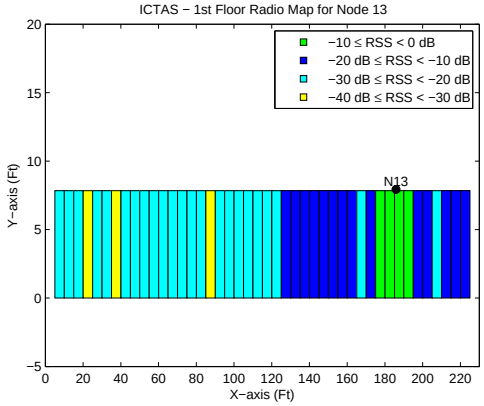

Figure E.3: Radio Map of Node N13

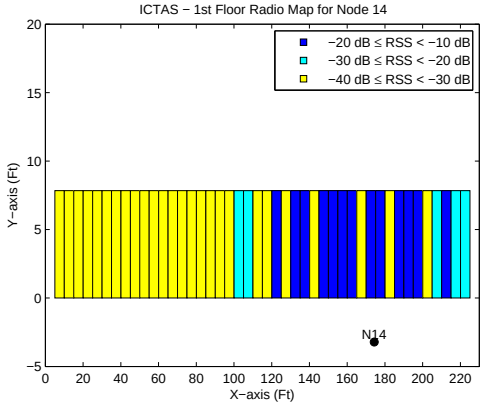

Figure E.4: Radio Map of Node N14

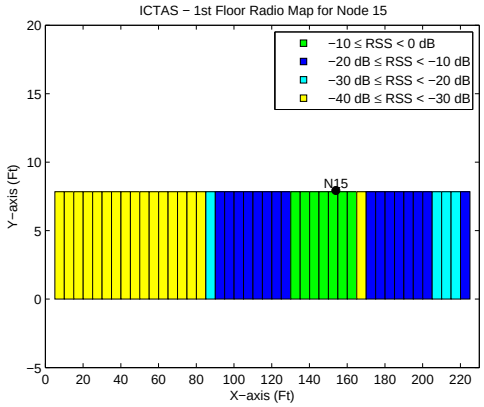

Figure E.5: Radio Map of Node N15

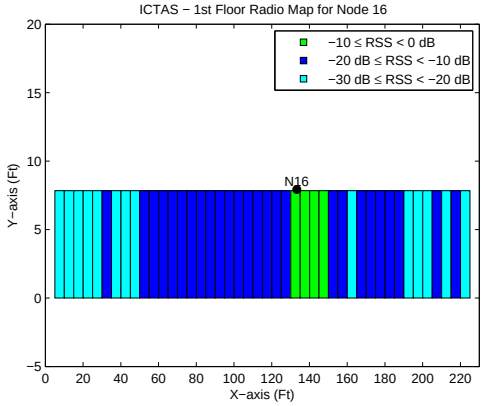

Figure E.6: Radio Map of Node N16

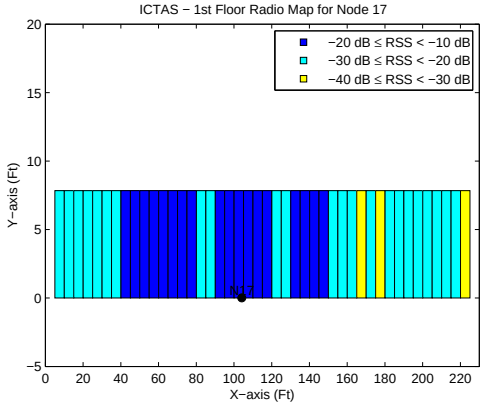

Figure E.7: Radio Map of Node N17

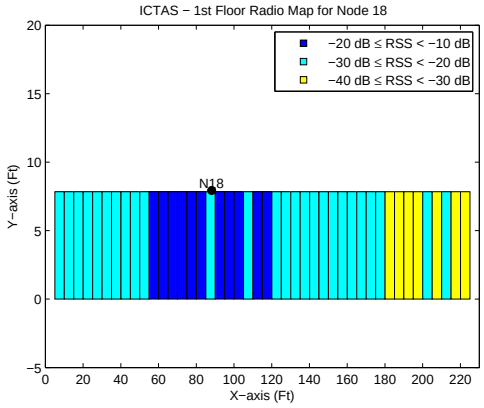

Figure E.8: Radio Map of Node N18

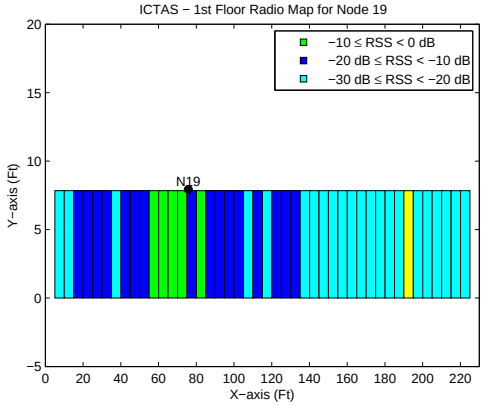

Figure E.9: Radio Map of Node N19

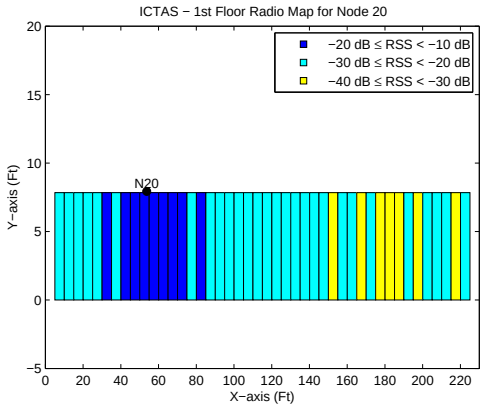

Figure E.10: Radio Map of Node N20

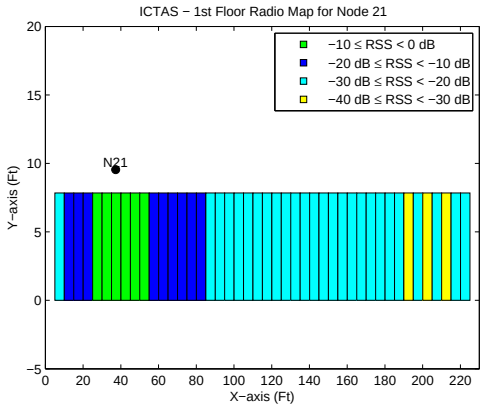

Figure E.11: Radio Map of Node N21

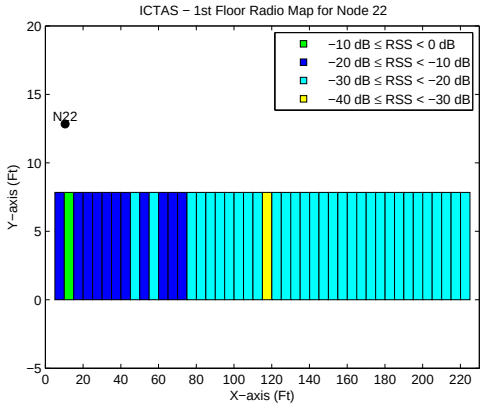

Figure E.12: Radio Map of Node N22