3-23-2018

RSS-based Device-free Passive Detection and Localization using Home Automation Network Radio Frequencies

Tiffany M. Phan

RSS-BASED DEVICE-FREE PASSIVE DETECTION AND LOCALIZATION USING HOME AUTOMATION NETWORK RADIO FREQUENCIES

THESIS

Tiffany M. Phan, 2d Lt, USAF

AFIT-ENG-MS-18-M-054

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-18-M-054

RSS-BASED DEVICE-FREE PASSIVE DETECTION AND LOCALIZATION USING HOME AUTOMATION NETWORK RADIO FREQUENCIES

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Cyber Operations

Tiffany M. Phan, B.S. Physics, B.A. Mathematics

2d Lt, USAF

March 22, 2018

AFIT-ENG-MS-18-M-054

RSS-BASED DEVICE-FREE PASSIVE DETECTION AND LOCALIZATION USING HOME AUTOMATION NETWORK RADIO FREQUENCIES

THESIS

Tiffany M. Phan, B.S. Physics, B.A. Mathematics
2d Lt, USAF

Committee Membership:

Dr. Richard K. Martin
Chair

Dr. Scott R. Graham
Member

Dr. Michael A. Temple
Member

Abstract

This research provided a proof of concept for a device-free passive (DfP) system capable of detecting and localizing a target through exploitation of a home automation network's radio frequency (RF) signals. The system was developed using Insteon devices with a 915 MHz center frequency. Without developer privileges, limitations of the Insteon technology like no intrinsic received signal strength (RSS) field and silent periods between messages were overcome by using software-defined radios to simulate Insteon devices capable of collecting and reporting RSS, and by creating a message generation script and implementing a calibrated filter threshold to reduce silent periods.

Evaluation of the system deployment in a simple room with no furniture produced detection rates up to P_D = 100% and false positive rates as low as P_F = 1.6% for baseline threshold detection along the line of sight (LOS) in a simple tripwire setup. Signal attenuation of foam blocks at different distances along this LOS ranged from 2.2-4.4 dB. Cell-based fingerprinting for localization using multiple nodes in this room achieved accuracy only as high as P_A = 5.4% and false positives only as low as P_F = 88.3%. A context-based localization method was developed in response and was able to achieve P_A = 28.3% and P_F = 40.0%. The system was then deployed in a similar room containing several metal objects and achieved P_A = 42.2% and P_F = 0.0%. Deployment in a similar room with RF absorbent objects achieved P_A = 23.3% and P_F = 53.3%.

Feasibility of exploiting RF of a home automation network for DfP indoor detection and localization was demonstrated. Despite not achieving optimal localization performance, the results showed promise for future DfP system deployment on top of home automation RF devices.

Table of Contents

Page

Page

List of Figures

Figure Page

List of Tables

Table Page

List of Abbreviations

ACK	acknowledgement
AoA	angle of arrival
API	Application Programming Interface
APK	Android Package
BLE	Bluetooth Low Energy
DCPS	data collection and processing system
DfP	device-free passive
GPS	global positioning system
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
IPS	indoor positioning system
JAR	Java Archive
LOS	line of sight
NAK	non-acknowledgement
OPS	outdoor positioning system
PL	power line
RF	radio frequency
RFID	radio frequency indentification
RMSE	root-mean-squared error
RSS	received signal strength
RSSI	received signal strength indicator
RTI	radio tomographic imaging

SDR software-defined radio

SRTI shadow-based radio tomographic imaging

TDoA time difference of arrival

ToA time of arrival

VRTI variance-based radio tomographic imaging

WSN wireless sensor network

URI Uniform Resource Indentifier

URL Uniform Resource Link

List of Symbols

Symbol		Page
$\overline{\overline{R}}_e$	Offline data matrix of calibrated $\overline{R}_e$	59
$\overline{R}_e(c)$	Offline data vector of all N $\overline{R}_e$ for a cell	59
$\overline{R}_e(c,n)$	Offline calibration value of $\overline{R}_e$ for a location from a node	59
c	A cell	59
n	A node	59
T	Time period of calibration	59
t_0	Start time in s of online collection period	59
τ	Time period in s for a localization event	59
$\overline{R}(t_1,n)$	Observed mean RSS from a node	59
t_1	Time in s at the beginning of a period for a localization event	59
$\overline{R}(t_1)$	Vector of all N observed mean RSS	59
c_p	Selected, most-probable cell location of target	60
c_{true}	True cell location of target	60
$\sigma_e(c,n)$	Standard deviation in dB from offline calibration of a cell from a node	60
$n_e(c,n)$	Number of steps recorded for offline calibration of a cell from a node	60
$\sigma(t_1,n)$	Standard deviation in dB from online observation from a node	60
$n_o(t_1,n)$	Number of steps recorded for online observation from a node	60
$P(c,n)$	Welch's p-value comparing $\overline{R}_e(c,n)$ and $\overline{R}(t_1,n)$)	61
P_A	Accuracy	61
D	Number of true positives	63
F	Number of false positives	63
$P_D(c)$	Cell true positive rate	63
$P_F(c)$	Cell false positive rate	63

Symbol		Page
D_H	Set of distances in ft from the Hub tested in detection	64
s	Silent period, or no 915 MHz signal	82
$\overline{R}_s$	Mean silent period RSS in dB	82
R_b	Single silent period RSS value in dB	82
$\overline{\max R}_s$	Average maximum R_s in dB observed in several collections	83
f_s	Silent period filter threshold in dB	83
g	Threshold requirement parameter of nodes for a candidate to have $P(c,n) > 0.05$	108
x	A context region	114
X	Total number of context regions	114
$\overline{R}_e(x,n)$	Mean RSS in dB of a region from a node	115
$\sigma_e(x,n)$	Standard deviation of the RSS for a region from a node	115
$n_e(x,n)$	Number of collection steps for RSS for a region from a node	115
x_{true}	True region of target location	115
x_p	Selected, most probable region of target location	115
$P(x,n)$	Welch's p-value comparing $\overline{R}_e(x,n)$ and $\overline{R}(t_1,n)$	116
$P_D(x)$	Context region true positive rate	117
$P_F(x)$	Context region false positive rate	117

RSS-BASED DEVICE-FREE PASSIVE DETECTION AND LOCALIZATION
USING HOME AUTOMATION NETWORK RADIO FREQUENCIES

I. Introduction

Recent years have seen Internet-of-Things (IoT) services become more prevalent in homes and other indoor settings. Many home automation devices use these services to improve functionality and to allow a better interface to devices for users. Since home automation networks are capable of changing and monitoring indoor environments, they can be exploited for indoor detection and localization of human targets. Detection and localization can additionally be performed without requiring targets to carry devices or intentionally participate in the process. Therefore, the research introduced in this chapter explores the exploitation of home automation networks to conduct device-free and passive indoor detection and localization.

1.1 Background

The development of systems for indoor detection and localization of a human subject is a popular topic of research. Detection is the indication of whether a human target is present in a monitored area. Localization is the determination of the position of a target in a monitored area. These topics are often also associated with tracking, or the monitoring of the trajectory of a human target. As opposed to outdoor methods, indoor methods focus on performing in enclosed settings with many obstructions.

Although better performance can be achieved using pre-positioned, dense networks and active target input, a more covert and cost-effective system is possible. A covert system works without drawing the target's attention to the system and without requiring the target to carry any devices or otherwise actively participate in the detection or localization process. Such a system is known to be device-free and passive (DfP). A cost-effective system also makes use of the devices already existing in the area of interest.

To achieve DfP detection and localization, an indoor system can exploit devices already capable of affecting changes to the indoor environment and of monitoring changes in the environment without requiring the target to carry any devices. One method to achieve this is to use signals already designed to permeate indoor structures, signals like those associated with a wireless sensor network (WSN).

A WSN is a group of nodes that collect data based on their specific functions and that are connected together to form a network. An indoor detection and localization system can exploit the devices and signals of a WSN to provide good performance while also reducing both cost and installation time. Additionally, most WSNs have low unit cost, so many nodes may be purchased and deployed in a large network that covers a larger indoor area than could feasibly be accomplished with wired security or surveillance systems [1]. An increasingly popular WSN used indoors is the home automation network.

Home automation technology has existed for decades, but with the recent surge in IoT services, the prevalence of home automation devices has increased as well. These devices often have wireless capabilities and communicate with other nodes and a central hub to provide automated functions in an indoor setting. Therefore, their signals and communication protocols are designed to function even through walls. Since home automation devices function well indoors, they are a prime candidate for exploitation for indoor detection and localization purposes.

1.2 Motivation for Research

A DfP system capable of indoor detection and localization has many potential applications. This includes a range of commercial and personal uses. For example, the system can be used in a mall by owners to monitor the flow of people visiting stores without necessarily purchasing items [2]. The location of assets and guides in stores can be determined without encroaching on customer privacy [3]. Automated, indoor location-based kiosks can give out safety information. The system can monitor hospital patients or elderly residents who may not be able to carry monitoring devices or whose privacy matters [4]. First responders can use the system to determine where a victim is located in a building when conditions prevent video surveillance from successfully capturing footage [5].

The exploitation of IoT home automation devices allows other applications as well. Home automation aspires towards completely automated responses for improving human life; users will be able to control appliances with completely passive input, that is, without the need for voice commands or other user input from mobile apps. This is the same principle of DfP detection and localization. Therefore, both home automation devices and a detection and localization system can work together to provide improved home automation control. Smart assistants can monitor targets and provide reminders or suggestions. Robot assistants can access the system to navigate a room to reach a target.

A popular system of human localization is the Global Positioning System, but while this system functions well outdoors when the line of sight between the satellite and the tracked device is clear, it struggles to achieve reliable performance indoors. Current research on indoor detection and localization systems is largely on device-based systems. Those that are DfP have not yet exploited a specific home automation technology to carry out detection and localization.

1.3 Statement of the Research Problem

The purpose of this research is to provide a proof of concept for a DfP system capable of detecting and localizing a target through the exploitation of a home automation network's RF signals.

The system is developed on an existing home automation network, and Insteon technology is the chosen home automation technology pursued in this work. Research on the technology is needed to understand how to use the network for detection and localization and to discover any limitations that the proof of concept will need to overcome. From this research, investigations need to be conducted to develop the tools for the exploitation of the Insteon devices. When the proof-of-concept system is developed, the system is then deployed and tested to evaluate the performance of the system for future development. Both detection and localization are tested.

Potential contributions of the research include demonstrating the feasibility of exploiting home automation networks for indoor detection and localization. The system should also be designed to minimize monetary costs from hardware and installation needs. Maintaining covertness is a concern. Other goals are to achieve high accuracy and low false positive rates during deployment and testing. Robustness, or consistent performance in different rooms, is also sought.

1.4 Document Organization

The remaining chapters are organized as follows. Chapter II provides background information regarding techniques and methods for human detection and localization. It also presents related works, compares home automation technologies, and explores Insteon devices. Chapter III presents the details of the methodology, including variables and definitions, for the testing of this proof-of-concept system. Chapter IV continues system development with investigations of the Insteon devices and develops algorithms

and tools for detection and localization. Chapter V conducts experiments on the deployed system and presents several results and analyses, including signal attenuation effects and other environmental effects. Chapter VI presents a summary of the research and provides recommendations for future research.

II. Background and Literature Review

This chapter presents background on the thesis topic of exploiting a home automation network for DfP detection and localization indoors. To understand the novelty of this research and the approach taken, several topics are first discussed. Section 2.1 presents different techniques for detection and localization and argues for a DfP technique. Section 2.2 compares different methods for DfP detection and localization. Section 2.3 summarizes related works in indoor detection and localization. Section 2.4 compares different home automation technologies for potential exploitation. Finally, Section 2.5 presents Insteon-specific details.

2.1 Techniques for Human Detection and Localization

This section presents different techniques of human detection and localization systems and compares them. A technique in this research is defined as how a system physically approaches the problem and is identified by three categorizations: outdoor vs. indoor, active vs. passive, and device-based vs. device-free. Since the focus of this research is a proof of concept for a single-target detection and tracking system over an already-existing WSN while minimizing costs and maintaining covertness, an indoor, passive, and device-free technique is ultimately selected.

2.1.1 Outdoor vs. Indoor.

Outdoor Positioning Systems.

Outdoor positioning systems (OPS) consist of technologies that include satellite, radar, or large radio systems. They are large-scale in that they can be used to track objects over a wide area, but primarily function only outdoors and have limiting high costs.

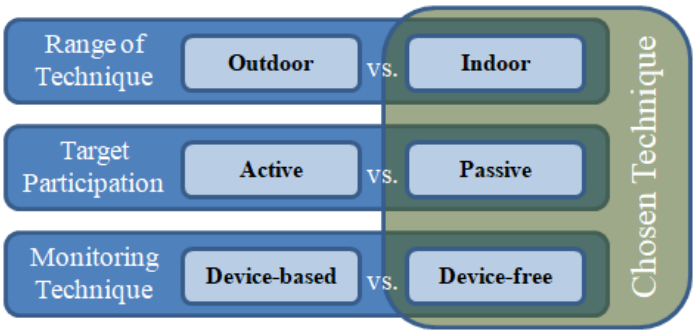


Figure 1. Diagram depicting discussed techniques with chosen technique highlighted

For example, global positioning systems (GPS) are capable of tracking devices through satellite communication. Though they once were limited to military-only use, consumer availability has made GPS the most popular human tracking technology. In fact, GPS is a feature in many everyday items for individual consumers, including smartphones, vehicle navigation, and fitness watches. However, GPS still has relatively high costs and consumes a lot of energy, which makes it difficult to be ubiquitously deployed in many nodes [2, 6].

The use of OPS has been wide-spread. However, the large scale of these systems means that their technologies are often restricted, resulting in large monetary costs for service usage or hardware and installation. In addition, with no functionality in indoor environments and limited performance in cities with tall building or between mountains due to required line of sight for communication to satellites, OPS precision is lacking [1].

Indoor Positioning Systems.

Indoor positioning systems (IPS) consist of technologies that function inside buildings and typically use signals and fields collected by small, electronic devices. IPS are not as popular as OPS, but research has increased in the past decade towards the creation and improvement of IPS.

IPS can be categorized into two types of technologies: non-radio technologies and radio-based technologies. Non-radio technologies, like cameras, can be expensive due to high equipment and installation costs, and often require unobstructed line of sight. Radio-based technologies, however, can be deployed in any small environment, and often leverage already-existing radio signals. In fact, radio-based technologies are often associated with home automation technologies, most of which operate on some radio frequency. In addition, radio-based technologies are more interesting because they can penetrate non-metal walls and do not necessarily require a clear line of sight like GPS does or extra equipment like floodlights do [5].

Innovation with currently-existing technologies for use in detection, positioning, and tracking keeps this area of research popular and continues to improve upon the faults of OPS. While some IPS may use heavy-duty installations of new, positioning-specific technology, research exists using currently-deployed technology for positioning [7]. For example, the ubiquity of Wi-Fi has encouraged researchers to deploy IPS relating to Wi-Fi signals. Whether it is a part of the main functions of the technology or an additional module that can be deployed on top of the technology, these innovations are changing the possibilities of indoor positioning.

2.1.2 Active vs. Passive.

The categorization of techniques into active or passive involves whether the technique requires active participation towards detection or localization. This categorization is often confused with that of device-based vs. device-free techniques, which have to do with objects. Some works, like ref. [8,9], group active and device-based techniques together because the target must actively remember to carry a device in order to be monitored by the system. However, an active technique in general involves conscious and intentional action from the target that enables detection or localization.

For example, suppose that Target A carries a communication-less tag whereas Target B carries a device computing and transmitting information to be used for tracking, even if it is only forwarded to another node. Then Target A is in a passive system and Target B is in an active system, because while Target A's tag does not actively participate, Target B's device does. Also, as a second example, if the target must knowingly interact with the environment, such as turning appliances on, for the monitoring to occur, then this is active participation. Hybrid approaches, using both active and passive targets, are also possible.

Active participation can be restrictive; not all individuals to be monitored are guaranteed to willingly participate. Passive participation, when the target does not have to actively recognize that they are participating, allows for greater application and use and provides system covertness.

2.1.3 Device-based vs. Device-free.

The question of whether the target is carrying a monitored node determines if the technique is device-based or device-free. For example, in the first example above, both Target A and Target B were in a device-based system. In the second example, the technique was device-free. While this may be straightforward, the concept of the success of a device-free technique may not be as apparent.

Device-based Techniques.

Before explaining the device-free technique, the device-based technique is discussed. The two "devices" that are typically used in a device-based technique are a mobile device, like a smartphone, or a tag, like an RFID tag [3]. Typically, the difference in the two is that the tag is passive, whereas the mobile device is active due to its larger functionality and capability for active participation by using its cameras and sensors.

For example, the technology behind GPS is device-based, because the target is actually the GPS-capable device communicating with global satellites. Another example is the light-sensing camera on a smartphone that may use triangulation based on the light density from indoor lights [3]. Using the accelerometer, gyroscope, and digital compass in a smartphone or monitoring Wi-Fi or Bluetooth signals emitted from the smartphone can also achieve active, device-based localization. The smartphone either communicates this information to monitoring nodes or some mobile app.

Tags, on the other hand, do not collect any data to be transmitted, but they may be involved in the communication. The tag can consist of ultra-wideband, infrared, radio-frequency identification (RFID), or other wireless sensors [3]. Signals are bounced off of the tag and monitored, but the tag itself does not participate in the exchange of information. For example, RFID are reference tags whose locations can be detected and tracked using a RFID reader.

Device-based approaches often give more precision and accuracy in localization because the target is carrying the device. However, this assumes that the target is carrying the device and that the device is fully equipped or properly functioning, e.g. no bent tag or drained battery. Additionally, device-based techniques may raise privacy concerns if the device is always worn by the target, who may want privacy in certain areas yet have multiple uses for the device.

Device-free Techniques.

Without any device on the target, a device-free technique relies on signals in the environment and the target's natural effect on these signals. Many of these signals are available and can be monitored by different techniques, including camera feed, infrared, ultrasonic, and radio frequencies [3].

A simple example of the device-free technique is using a camera feed. The target

does not need to carry any devices, but instead is detected on the visual spectrum that cameras capture. Infrared works the same way; it exploits how the human body's heat signatures differ from the environment. Echolocation, based on measuring the time that sound takes to propagate and then reflect back to the source, is another simple example.

More complex is the use of non-visual and non-audio signals for detection and localization. Radio-based device-free techniques commonly exploit path loss and the human body's natural attenuation or reflection of signals to determine detection or location. For example, just like with echolocation, if a radio signal is sent out with the expectation that is it returned but after some time is not, then a system may consider that a detection event because the time of flight exceeded what was expected.

Device-free techniques are useful because they do not rely on the target to carry any devices, which reduces hardware costs and allows detection and localization to occur without needing the target to carry a device.

Summary.

Indoor techniques are interesting because they are able to exploit existing WSN signals, which could cut costs. Radio-based techniques favorably do not require unobstructed line of sight to achieve target monitoring. Passive techniques do not require active participation by the target, so can be covert. Similarly, device-based techniques are less obtrusive than the device-based one, which may be more precise yet is susceptible to device failures (e.g. forgetting the device or a drained battery).

An indoor, DfP technique then has low costs, performs covertly (without need of a carried device or active participation), and can be deployed indoors on any existing WSN. In accordance with ref. [10], the DfP approach should be used in an installed WSN to detect changes in the environment and monitor targets passively without requiring any devices to be attached to these targets.

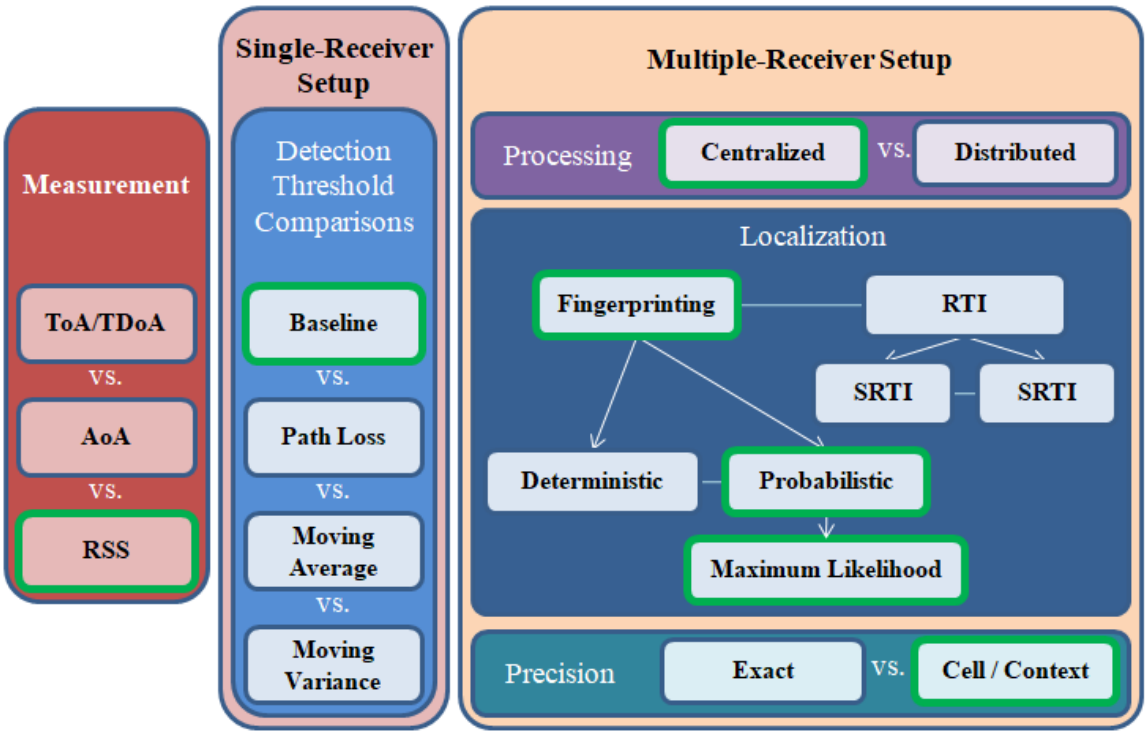


Figure 2. Diagram depicting discussed methods (measurements and approaches to detection or localization) with selected methods outlined in green

2.2 Methods for DfP Detection and Localization

This research aims to choose a method for DfP detection and localization that best fits these qualities: low cost, high precision and accuracy, and low computational overhead. Therefore, this section presents different methods that are used in DfP works. Methods consist of the measurement taken and the approach towards detection and localization. After a measurement is selected, DfP approaches for a single-receiver setup and those for a multiple-receivers setup are then presented for potential use.

2.2.1 Measurements.

Many factors affect the localization algorithm, including network architecture, sensor density, and geometric shape of the measurement area, but the most important factors in accuracy are the type of measurement and precision that measurement pro-

vides [2]. The measurements frequently used in DfP techniques are time, angle, and signal strength [1, 6, 11]. This section discusses each of these three measurements before arguing for the use of the received signal strength.

Time of Arrival / Time of Flight.

The measurement of time can be used in many different ways depending on how it may be measured at the nodes. For example, time of arrival (ToA) can be recorded as a one-way or round-trip measurement. The time of flight, or time difference of arrival (TDoA), is the difference between arrival times of a transmitting signal at two separate receivers [2]. Ref. [12, 13] are device-based works that used ToA or TDoA. GPS is an example of a TDoA system.

ToA and TDoA approaches can achieve high accuracy, but require strict time synchronization of the network, making its use costly [14]. In fact, synchronization error and multipath effects can reduce the accuracy of time-based measurements [2].

Angle of Arrival / Direction of Arrival.

Angle of arrival (AoA) is a measurement used to calculate the angle, either as amplitude or as phase, at which the signal arrives from one node to another [2]. AoA often requires additional hardware to calculate the relative orientation based on the measured arrival direction, thereby increasing implementation costs and power consumption [14]. Accuracy is affected by shadowing and multipath and its use has a limited interest to large antenna arrays [2]. WSNs with small sensor nodes cannot support this non-energy-efficient approach. The infrared-based work in ref. [15] and the device-based work in ref. [16] both used AoA measurements.

Received Signal Strength.

Received signal strength (RSS) is a measurement that takes into account the signal strength of a received signal. This is often measured in dB or some variation.

One challenge of using RSS is the unpredictability due to multipath fading [1]. Multipath fading is often due to environmental attributes causing reflections, but noise may also interfere with RSS. Ref. [17] claimed that their experimental study showed that RSS cannot be used as a metric for distance measurements in localization algorithms because of its inconsistent behavior and unreliable results at the extremities of its range. However, RSS can be processed before use in distance calculation [14].

RSS is favored for its simplicity and because it is often available as a recorded parameter in the receiver or in the communication message packets, which makes this measurement approach fast and cost-effective [5, 14]. It is also favored because of its low hardware configuration requirements [6]. The concept of path loss for signal strength is also a basis for many techniques.

Comparison.

AoA requires specified hardware, which is not often found in existing WSNs. Therefore, using AoA would increase hardware costs. RSS, on the other hand, is often available as a recorded parameter. As a result, using RSS would not accrue these additional costs.

Time can have high accuracy, but ref. [12] compared RSS and TDoA measurement approaches on localization and determined that while TDoA works better than RSS, there are locations where it does not produce good estimates, whereas RSS can work everywhere. This is because TDoA requires a clear connection with unobstructed line of sight, but RSS can overcome effects from obstacles and multipath.

2.2.2 Approaches for a Single Receiver Node.

The simple case of DfP techniques is detection using one receiver node and one transmitter node. This is also referred to as the "tripwire case". Detection is measured as the ability to indicate whether a human target is present in a monitored area. With RSS measurements, this is determined by monitoring RSS changes in the environment. There are a few ways to approach this simple-case problem, such as baseline signal comparison, path loss model comparison, moving average threshold, or moving variance threshold, which are all discussed below.

Localization is the technique used to determine the position of a target in a monitored area. (This is not to be confused with tracking, which follows the target's position over time.) Localization is a more complex problem than detection. However, it is associated with poor results when used in a simple setup like the tripwire case, so this is discussed more in the next section with the multiple-receiver setup.

Baseline Comparison.

Baseline comparison is an intuitive method for RSS-based detection where the target is detected in the area of interest if the RSS decreases significantly from the baseline. The level of significance may depend on several factors, and so configuration may be required, especially in large spaces or spaces with large multipath effects.

With just one transmitter, one receiver, and no obstacles, human presence can be detected when the observed RSS value decreases significantly from the baseline. Significance can be set based on expected values (e.g. using commonly known values for certain frequencies) or can be calibrated from previous test runs. The issue with using expected values is that it does not consider path loss or multipath effects, so these values are not universal and may result in lower probability of detection or in higher false positives.

Path Loss Threshold Comparison.

Detection along or near the line of sight can be obtained using the path loss model. The path loss model is

$$P_L(d) = P_L(d_0) + 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \tag{1}$$

where P_L is the path loss, d is the distance, d_0 is the reference distance, γ is the path loss exponent, and X_σ is the zero-mean Gaussian random variable associated with path loss due to shadowing.

With some points of calibration, the path loss exponent and shadowing variable can be calibrated. Then, the path loss model can be used to determine a threshold of RSS values that indicate that the target is in a subset of the area of interest, near the line of sight of the transmitter and receiver.

Considerations for when the target completely blocks the transmitting node, preventing most of the signal from reaching the receiver, or for when the target is far off the line of sight are necessary when using this method. This approach also requires recalibration for any significant changes in setup or environment. With more than one receiving node, this approach can also be adapted for localization.

Moving Average Threshold Comparison.

A statistical approach that compares two moving averages of the RSS of a single stream can remove the need for recalibration and was presented in ref. [10]. Let q_i be the RSS of a signal. The long and short moving averages of a single stream ($\alpha_{l,k}$ and $\alpha_{s,k}$, respectively) are simultaneously taken of a long window, w_l, and a short window, w_s, and compared (4) to a threshold, τ. If there is a significant change between $\alpha_{l,k}$ and $\alpha_{s,k}$, then a detection event occurred at time $t = k + w_s$, where k is the time index.

$$\alpha_{l,k} = \frac{1}{w_l} \sum_{i=k}^{k+w_l-1} q_i \tag{2}$$

$$\alpha_{s,k} = \frac{1}{w_s} \sum_{i=k+w_l}^{k+w_l+w_s-1} q_i \tag{3}$$

$$\left| \frac{\alpha_{l,k} - \alpha_{s,k}}{\alpha_{l,k}} \right| \underset{H_0}{\overset{H_1}{\gtrless}} \tau \tag{4}$$

where H_1 is the event of detection and H_0 is no event.

Moving Variance Threshold Comparison.

Similar to the moving average approach, the moving variance approach compares moving variance of the raw data, v_t, to the variance of the silent period, $\overline{v}_t$. This is also presented by ref. [10]. Unlike the moving average approach, the silent period must be calibrated beforehand for some period $[t_{start}, t_{end}]$. Detection occurs when the variance of the raw data stream is greater than the variance of the static period, or when $v_t > \overline{v}_t + r\sigma_v$, where r is the number of standard deviations, σ_v, that acts as the threshold.

$$\overline{q}_t = \frac{1}{w} \sum_{i=k}^{k+w-1} q_i \tag{5}$$

$$v_t = \frac{1}{w-1} \sum_{i=k}^{k+w-1} (q_i - \overline{q}_t)^2 \tag{6}$$

$$\overline{v}_t = \frac{1}{t_{end} - t_{start} + 1} \sum_{t=t_{start}}^{t_{end}} v_t \tag{7}$$

$$\sigma_v = \sqrt{\frac{1}{w-1} \sum_{t=t_{start}}^{t_{end}} (v_t - \overline{v}_t)^2} \tag{8}$$

Comparison.

Both the moving average and moving variance approaches can be performed in real time. Adding an additional constraint for detection like the number of concurrent alerts raised in a time period can also reduce the number of false positives [10]. In fact, ref. [10] was able to find 100% probability of detection with zero false positives with specific parameter values. However, these approaches require more computational overhead that may be too costly for a simple approach or goal. Instead, the baseline comparison should be sufficient with its low complexity.

2.2.3 Approaches for Multiple Receiver Nodes.

Similar to detection, there are multiple approaches to RSS-based localization. Two of the most-used approaches are fingerprinting and imaging, both of which are discussed below. Before presenting these approaches, a general question (centralized vs. distributed processing) of the multiple receiver approach is discussed.

Centralized vs. Distributed Processing.

In cases with just one transmitter and one receiver, the issue of centralized or distributed analysis approach is not of concern. However, these one/one scenarios are impractical, have coverage limitations and will result in more error for localization. With more than one receiving node, whether the nodes communicate to estimate distance is a significant question.

When the nodes communicate with each other, they use a centralized approach; when they calculate their estimations independently, they use a decentralized approach. Centralized algorithms are more accurate, but are not suitable for extremely large networks and have higher computational complexities and unreliability due to inaccurate accumulated information [2]. Decentralized algorithms, on the other hand, are more

computationally efficient and are easily implemented in large networks, but have design complexity if not synchronized, and individual nodes often have limited processing capability prohibiting computation [2].

A limiting factor in this research is that the nodes to be used should be part of an already-existing network, meaning that their computational capabilities are small or are limited to their intended function. As a result, the centralized approach is preferred for this research.

Radio Fingerprinting.

Fingerprinting is a classification-based approach to localization (that can also be used for detection). These methods identify patterns using a unique signature [8]. In terms of RSS-based localization, classification through fingerprinting consists of having a radio map containing measurements that change depending on where a target is and using this map to compare the observed readings [1]. Thus, this approach can be divided into two phases: the offline phase and the online phase.

The offline, or training, phase first involves choosing location points in the area of interest. This can be evenly-spaced in a systematic, grid-like fashion or can be irregularly-chosen points spread throughout the area of interest. Points away from obstacles may be preferred, as may points at locations of interest, such as near a bed, or near a certain kiosk. Regardless of how this map is defined, the next step is to calibrate the map. Calibrating the radio map consists of recording RSS values at each of the points chosen in the first step. These values should be an aggregate of values from all relevant receiving nodes. However, how these values are stored may differ [18].

The online, or observation, phase is the fingerprinting of the currently collected raw data against the data stored in the trained radio map. This can be approached many ways. For example, ref. [8, 19] used MATLAB® Naive Bayes and TreeBagger classifiers.

Ref. [18] compared probabilistic techniques to a deterministic technique and found that probabilistic techniques outperformed the deterministic one. The research also found that non-parametric algorithm approaches for estimating the density function outperformed parametric algorithm approaches.

The collection of unique signatures at specific locations allows for greater accuracy [2]. However, fingerprinting often requires "extensive calibration to build a signal fingerprint" and recalibration is required when the environment changes, such as when a piece of furniture is moved [20].

Ref. [10] performed localization by first constructing a passive radio map and then using a Bayesian inversion-based inference algorithm to classify the observed data as similar to the saved data at a point on the map. The algorithm assumes that all locations are equally probable, and its accuracy and precision depend on the number and location of the calibration points. In the offline phase, the map is constructed and the target stands at different points in the area of interest. The RSS values for each receiving node (n_i, for some $i \in \{1,2,...,N\}$, the set of all nodes in the system) are then recorded and stored as a vector, $\overline{s}_l$ at each location point, l. In the online phase, a signal strength vector, $\overline{s}_u$, of an unknown location is recorded and contains RSS values from all n_i. Then to determine the location of the target, the approach finds the location where the maximum probability occurs.

$$\arg\max_l(P(u = l|\overline{s}_u)) = \arg\max_l(P(\overline{s}_u = \overline{s}_l|l)) \tag{9}$$

Ref. [9] presented a cell-based fingerprinting approach for tracking. The room was first divided into equally-sized cells. Then, in the offline phase, 100 RSS measurements were recorded for each cell with the target having different orientations and positions within the cell. In the online phase, the median RSS from the training section was compared to the observed value and maximum likelihood was used, similar to (9). This

method does not have exact precision since the cells are of some area. However, this is not an issue for most applications that do not need precision beyond a context, like proximity to a bed or doorway, and this approach reduces calibration overhead [21].

The above fingerprinting approaches can only choose locations from the set of points calibrated in the offline phase, but in-between points can be computed using some other algorithm. Ref. [22] proposed spatial averaging, a deterministic, center-of-mass approach that allowed the estimated location to not be a predetermined point on the radio map, thereby making a radio map with continuous space. The task of adapting this for a probabilistic approach could be undertaken in order to improve localization accuracy without requiring more calibration overhead in the offline phase.

Radio Tomographic Imaging.

Radio frequencies can be used to build a tomographic image of the area of interest, and the image can then be analyzed to detect or locate a target. Radio Tomographic Imaging (RTI) is the "estimation from measurements along different spatial filters through a medium" [5].

This estimation is often done with propagation models, which have less calibration overhead than fingerprinting, but have demonstrated low accuracy [20]. Ref. [14] is an example of a work that chooses a model of the signal propagation path for localization instead of the fingerprinting method due to the time cost of calibration. Without needing to re-calibrate the system to update the map of the area of interest, RTI is capable of long-term localization, especially when paired with a state machine algorithm [23].

However, RTI often requires a lot of nodes to achieve better accuracy. This is because RTI requires line of sight to perform localization [22]. Without line of sight, such as from obstacles, accuracy degrades significantly [22]. The RTI experiments in the works discussed in Section 2.3 used 16 to 34 nodes, even in small areas of interest. This need

for more nodes increases hardware and installation costs and limits the coverage area possible to the number of nodes deployed [22].

Other basic approaches to RTI include the shadow-based RTI (SRTI) that estimates using mean RSS and the variance-based RTI (VRTI) that exploits the RSS variance during target movement [24]. In a comparison of SRTI, VRTI, and fingerprinting approaches, ref. [24] noted that SRTI and VRTI were both flexible but showed less accuracy than the fingerprinting method. Additionally, VRTI cannot detect stationary targets.

Some filters used in developing the image are the Kalman filter [6] or the bootstrap particle filter, the multiple particle filter, the Markov Chain Monte Carlo filter, and the Additive Likelihood Moment filter compared in [23].

Comparison of Multiple-Receivers Approaches.

The centralized-node communication approach is not scalable, has computational complexities, and can be unreliable, but often results in better accuracy. The decentralized approach is scalable and computationally efficient, but is limited by individual node processing power and synchronization.

Multiple receivers are better-suited for accurate localization. The classification-based fingerprinting method and the radio tomographic imaging approach were both discussed. While RTI requires less calibration overhead than fingerprinting and can achieve better accuracy, it is limited by the number of nodes that can be deployed in the network. Fingerprinting also does not require line of sight and permits some obstruction since it mitigates multipath effects [22].

Probabilistic fingerprinting approaches outperform deterministic approaches. The fingerprinting map is often created with discrete points, making accuracy dependent on the number of points calibrated. However, most applications may be satisfied with cell- or context-based precision.

Summary.

Measuring RSS is preferred for DfP techniques because it is easy to measure and exists for all nodes that send signals. This limits costs and allows an RSS-based DfP approach to be deployed on almost any exploited WSN.

For RSS-based, one-receiver, DfP detection, the baseline signal comparison approach is selected for its simplicity. Single-receiver DfP localization is not practical for accuracy, so a multiple-receivers setup is preferred despite additional design and computational complexities. For RSS-based, multiple-receiver DfP localization, the probabilistic fingerprinting approach can be used for smaller networks. Cell- or context-based precision is sufficient for most applications.

Given the goal of this research to exploit an already-existing WSN infrastructure's nodes for DfP detection and localization, a centralized approach is preferred over a decentralized one. The centralized approach allows all receiving nodes to share information, thereby increasing accuracy.

2.3 Related Works

Works Using Different Measurements.

Three works included non-RSS measurements in their experimental studies, all with varying technologies. Ref. [16] was an energy-based, Wi-Fi, device-based approach to localization using direct path signal through path loss comparison and AoA phase difference. Their method reduced distance error compared to RSS, but that performance was poor at a few locations, especially for weak links. For tracking, their method had errors on average of 5 m but found errors greater than 10 m at a few locations with one access point, but only 2.7 m for a system with five access points.

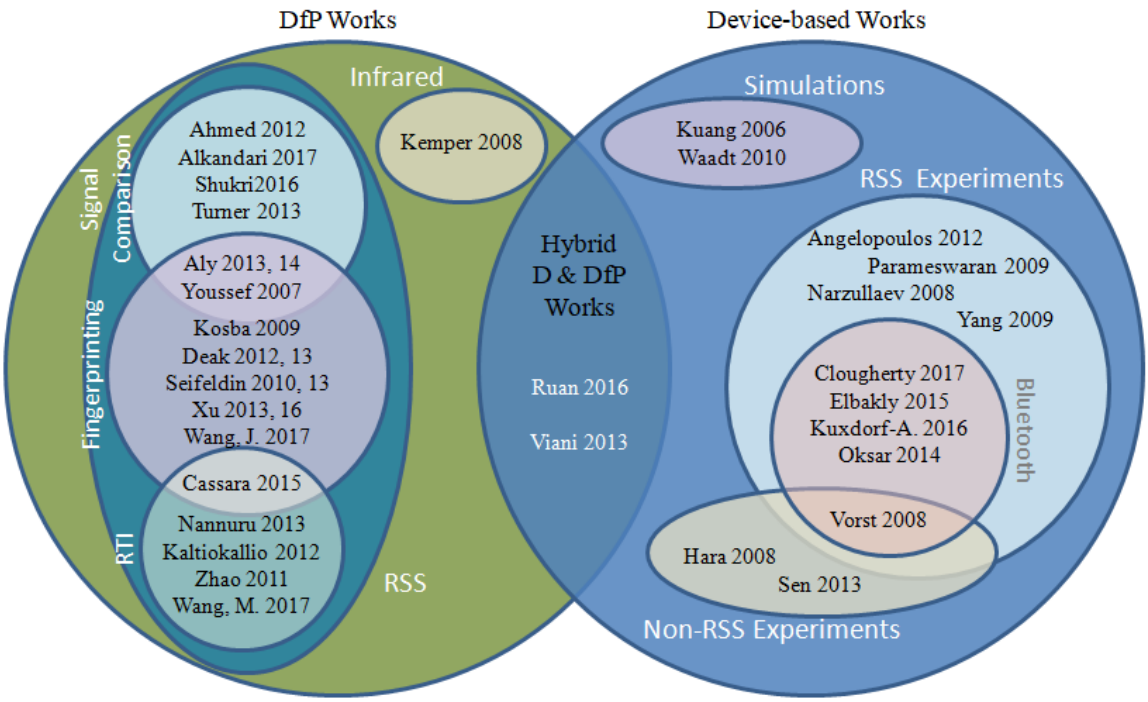


Figure 3. Diagram depicting studies of different related works reviewed

The other two works compared measurement approaches. Ref. [13] compared device-based techniques using RFID detection rate, Bluetooth technology RSS, or Wi-Fi technology ToA. They found that RFID and Bluetooth are more accurate than Wi-Fi with median errors of 0.435 m, 0.474 m, and 3.545 m, respectively. Ref. [12] compared RSS maximum likelihood and TDoA least squares approaches for a DfP system using IEEE 802.15.4 technology attached mostly to the ceiling. They concluded that RSS was more robust and that TDoA was more accurate.

Device-Based Bluetooth Works.

In addition to ref. [13], four other reviewed works specifically used Bluetooth technology, and all were device-based techniques. Ref. [25] tested the possibility of localization using smartphones measuring RSS from Bluetooth Low Energy (BLE) beacons and confirmed that the beacons "provide micro-location data that increases accuracy and

functionality in smart home technologies," but noted several limitations with a device-based system. Ref. [26] introduced an RTI system described as a calibration-free fingerprinting method, and found that the system was robust to temporal changes and could provide up to 187% improvement to traditional fingerprinting methods. Ref. [14] used RSS and triangulation and found that the user's real position could be estimated up to 30 cm without averaging and smoothing routines; with, the maximum deviation was reduced to less than 15 cm. Ref. [27] used a minimum error function (10) to estimate the location of users carrying Bluetooth-transmitting devices and found that their method resulted in a root-mean-square error (RMSE) of 2.309 m, but capitulated that this could be improved upon with optimal base station placement and minimizing reflections and path loss from walls and furniture.

$$\mathrm{EF} = \sqrt{\left(\frac{(d_1/d_2)^2}{S_2/S_1} - 1\right)^2 + \left(\frac{(d_1/d_3)^2}{S_3/S_1} - 1\right)^2} \tag{10}$$

where S_1, S_2, and S_3 are received signal strengths of base transmitters located d_1, d_2, and d_3 away from a transmitter.

Other Device-Based & Hybrid Works.

Unless otherwise stated, the following are device-based works. Interestingly, ref. [17] investigated the reliability of RSS as a device-free measurement and concluded that it was neither reliable nor consistent for distance measurements in localization algorithms. However, no other literature occurred in research that corroborated these findings. Instead, most device-based works often found success.

Ref. [28] used a moving mote that would send localization messages to anchor motes. The researchers found greater than 95% success in two of the four quadrants of a room with a new device-based localization algorithm and just four IEEE 802.15.4 anchors. Most other works improved upon existing methods or compared them.

Ref. [20] proposed a modified one-slope calibration method for device-based location prediction on a Wi-Fi network and were able to predict RSS with higher accuracy than a normal one-slop or multi-wall model. Ref. [29] improved RSS-based lateration techniques in Wi-Fi and Zigbee networks and found that their regression method in simulation and experiment significantly improved localization accuracy for RSS-based lateration, with median error improved over 29% in both networks, and maximum localization error improved over 50%.

Two works used only simulations to compare algorithms. Ref. [30] used a direct triangulation algorithm and a maximum likelihood estimator (MLE) algorithm on plume intensity, which is similar to RSS, and compared them when the number of sensors were varied and when the level of noise was varied. They found that neither algorithm performed well when there were fewer sensors due to noise, but the direct triangulation algorithm was preferred for its simplicity so long as the sensors are not in a line and are close to the source to minimize the effects of noise. When varying the background noise, they found that the MLE algorithm is more robust. Ref. [31] compared trilateration of a mobile node transmitting RSS using maximum likelihood estimators, observation of RSS path loss using least-squares, and RSS fingerprinting. They found that localization accuracy depends on the number of anchor nodes and on the target node's distance from the network's center. The maximum likelihood and fingerprinting estimators outperformed the least squares estimator and showed similar performance, but maximum likelihood's lower complexity was recommended.

Other comparisons: ref. [32] compared multi-lateration, min-max, and fingerprinting algorithms for Zigbee devices transmitting RSS and found that while all resulted in an average error of less than 60 cm, multi-lateration gave better results than min-max and with more reference nodes, fingerprinting had better results. Ref. [33] found that the maximum likelihood algorithm in device-based approaches yielded better performance

compared to the min-max algorithm when the number of nodes were higher because it weights the RSS, and that multi-lateration fared worse than either of these algorithms. Ref. [34] compared two Euclidean-distance, RSS approaches on Zigbee nodes and found that the fingerprinting technique resulted in better accuracy compared to the maximum likelihood algorithm (0.5 m vs. 1.2 m).

Two works used a hybrid approach of active and passive systems. Ref. [4] proposed a system that leveraged human-object interactions to affect the RF field and were able to achieve 95% localization accuracy and 58 cm tracking error for a single person in multiple rooms. However, issues included the impracticality of installing sensors on all appliances. Ref. [7] wanted to exploit the entirety of the home wireless network for localization and activity monitoring. In their device-based technique, smartphones were carried and an app monitored RSS data for real-time position estimation with mean localization error of 3.24 m with only three access points. In their device-free technique, they monitored appliances in the home and determined real-time location estimation from power usage and RSS with mean localization error of 3.61 m.

DfP Infrared Work.

Only one reviewed work used infrared technology for DfP indoor localization. With three sensors and a least-squares triangulation algorithm, ref. [15] measured a maximum error of 80 cm but were confident that an additional ±20 cm of accuracy was possible with optimized sensor configuration.

RSS DfP Signal Comparison Works.

Seven of the reviewed works used signal comparison for DfP detection and/or tracking using a one-transmitter, one-receiver setup in a room or smaller space. Of these, only ref. [35] used Wi-Fi technology; the others used IEEE 802.15.4 technology. Ref.

[11,35,36] measured RSS and graphed it as a function of time to record fluctuations due to human presence, whereas ref. [37] graphed the observed frequencies of RSS. Works using signal comparison appear to do so to verify that DfP human detection and tracking is possible or to test effects.

For example, ref. [36] showed that a slow walking pace significantly reduces RSS, but fast paces only do so slightly, and that greater numbers of humans reduce the signal strength more. Ref. [11] found that human attenuation in their system was approximately 3.97 dBm on the line-of-sight (LOS) path, that away from the LOS the presence could be adequately detected up to 1.0 m away, and that human movement caused RSS fluctuations and variations of around 10-15 dBm. Ref. [37] found that even without human movement, signal strength and its fluctuation are different at varying antenna angles, and that human movement from slower to faster paces increased the fluctuations in RSS. Ref. [35] showed that higher accuracy is found with longer time duration when using detection to estimate speed (finding mean accuracy of 84.47% and 94.41% for unidirectional and round-trip traversals of a straight-line path). Ref. [38,39] concluded that wall-mounted access points (for Wi-Fi) resulted in higher localization accuracy when compared to ceiling-mounted ones, that higher frequency (5.7 GHz vs. 2.4 GHz) resulted in lower accuracy, and that people moving outside the area of interest do not affect the data streams in the area of interest.

Ref. [10] found 100% probability of detection with zero false positives using moving average and moving variance approaches with a two-transmitter, two-receiver setup.

RSS DfP Fingerprinting Works.

Fingerprinting was popular among the reviewed works. Most approaches (ref. [10, 18, 22, 40, 41]) leveraged Wi-Fi technology, and while ref. [8, 19] used only IEEE 802.15.4 technology, ref. [7] did not discriminate between the two 2.4 GHz frequencies. However,

ref. [9, 21] used technology on the 433.1 MHz and 909.1 MHz frequencies. Experimental setups were either a single room or a single floor, and the number of nodes were as few as two transmitters and two receivers to as many as sixteen total.

Errors were less than 3.6% for the three MATLAB® classifiers (Naive Bayes Gaussian, Naive Bayes Kernel, and TreeBagger) used in ref. [8, 19]. The Bayesian inversion inference algorithm in ref. [10] resulted in a localization accuracy between 86.3% and 89.7%. The wavelet feature-isolating approach in ref. [40] found simultaneous localization and activity recognition accuracy of 95.4% and 59.3% for one and two targets, respectively.

Ref. [9, 21] both presented a cell-based, probabilistic fingerprinting approach using linear discriminant analysis that achieved 97% cell estimation accuracy with 0.36 m localization error distance for one person. In ref. [9], they further found that their approach maintained an accuracy of over 90% with a substantial reduction in number of devices (sixteen to eight) and with fewer calibration samples (100 to sixteen per cell).

The deterministic (Euclidean distance with post-processing spatial and temporal averaging) approach in ref. [22, 41] enhanced accuracy to give a median distance error of 6.74 m and a 100% probability of detection in typical environments. However, ref. [18] compared four fingerprinting algorithms and concluded that non-parametric probabilistic algorithms outperformed others. They also concluded that the configuration of nodes and the radio map data most affected accuracy.

Ref. [24] compared fingerprinting with RTI and found that their fingerprinting system had 0.8 m accuracy and could detect a still target, but had no flexibility and was expensive to configure.

RSS DfP Radio Tomographic Imaging Works.

RTI works often deployed many nodes, with the reviewed works using as many as 34 nodes (ref. [42]). Ref. [23] used simulations to show that the RMSE increased when

the number of nodes removed from the system was increased, but ref. [24] found that as few as sixteen nodes could still maintain DfP usability. Most works used IEEE 802.15.4 technology, but ref. [43] used Wi-Fi technology.

Ref. [44] demonstrated successful tracking of multiple human objects was possible using the exponential model on mean attenuation of the RF signal of a link by a target, a modified skew-Laplace measurement model, or a proposed magnitude measurement model; in three different sites of different multipath effects but still small rooms, the researchers successfully tracked up to three targets using all models.

Long-term monitoring is possible with RTI, as was shown in ref. [23], who presented a system that used a radio map based on RTI with a state machine of some areas of interest in a single-floor, single-bedroom apartment over a week. It was able to avoid recalibration of the system, and they found only 0.23 m RMSE of the position.

Ref. [24] compared the RSS-based DfP localization algorithms SRTI, VRTI, and classification and determined that each algorithm had their own strengths and weaknesses, so none could be preferred in the general case. SRTI had accuracy of 2 m, was capable of automated configuration, could detect a still target, and was flexible. VRTI had an accuracy of 1.2 m, was also flexible, and did not need configuration, but could not detect a still target.

Some works tried to improve upon traditional RTI methods: ref. [43] introduced an adaptive weighting RTI to improve accuracy in Wi-Fi environments with multipath and interference and found that their method outperformed the traditional RTI method by 4.51% in localization accuracy with interference. With greater interference, the adaptive weighting method showed even better performance. Ref. [42] proposed SubVRT, a subspace decomposition method to reduce noise for variance-based DfP localization and were able to reduce localization RMSE by 41% (0.74 m vs. 1.26 m) compared to a normal variance-based approach. They did so by removing intrinsic motion's spatial signature

after finding that intrinsic motion (e.g. moving leaves) resulted in higher RSS variance than extrinsic motion and considered the motion to be noise in the system.

Summary.

The current focus of indoor localization research largely seems to be to improve upon existing methods and to choose the best method. However, home automation networks have not been explicitly exploited. Furthermore, most of the experimental setups are different from each other, making it difficult to choose a method in indoor localization. By exploiting a home automation network though, the DfP setup can be more generalized since many home automation needs and capabilities are similar from home to home. Fingerprinting requires calibration and works with even small numbers of nodes, which is more likely in home automation scenarios.

Different measurements are possible for indoor localization, but when compared, RSS was more robust and did not require a clear LOS. Despite claims that device-based methods could estimate location better than device-free methods, the accuracy and errors reported from both sides are confounded by other variables and do not indicate favor either way. Still, device-free methods are mostly passive, which this research prefers.

None of the reviewed works used devices specifically intended for home automation. The majority of systems involved frequency bands at 2.4 GHz due to the resonance frequency of water, but 915 MHz signal attenuation also occurs. The IEEE 802.15.4 technology is most popular, with Wi-Fi next. Only four works exploited sub-1 GHz frequencies, but a few works tried to eliminate interference in the 2.4 GHz band.

Between fingerprinting and RTI for localization, fingerprinting could see significant inaccuracies in estimation, but by taking a cell-based approach, this inaccuracy is reduced. With home-based human detection and localization cases, it is unlikely that more precise localization is required.

2.4 Comparison of Home Automation Technologies

Home automation technology is of interest in this research for several reasons. Foremost, DfP techniques can exploit the already-existing network that most home automation technologies provide. This eliminates many costs that are associated with a system that is specified for localization only.

With continuing improvements in support and availability, homeowners and other building designers are deploying home automation networks throughout the country. This increased use and popularity of home automation technology also increases the relevance of a DfP system that exploits the home automation network.

This section considers several home automation technologies that may be exploited for DfP purposes. While many of these technologies can be used for DfP detection and localization, the technology used in this research must be radio-based, low-cost, and easy to exploit.

Lone Sensors.

Home automation technology has many forms and functions. In the case of lone sensors, this general technology is used in home automation networks but does not necessarily comprise its own network. For example, house alarm systems use a sensor that, if triggered, sends information to a monitoring console. Their function is limited to sensing a single event and sending it to another device. Outside of this simple conditional function, they provide no other use, making their deployment small. This makes exploiting lone sensors impractical. Motion sensors are similarly low-cost and have low power consumption, yet are often multi-functioning with their own processing and communication components. However, they can often only locate moving objects, not stationary ones, and purchasing detection-specific devices is not desired.

Surveillance Cameras.

Surveillance cameras violate user privacy since they capture too much information about the target. The data stream is large, making computational costs for processing recorded footage high. Line of sight is also required. Furthermore, while multiple cameras may be connected in a network, this is in the realm of a separate technology, e.g. Wi-Fi, commonly.

Wi-Fi.

This wireless technology is based on the IEEE 802.11 standards and is popular in home WSNs used primarily to connect devices to the Internet. Many home automation devices may use Wi-Fi technology to communicate to a home network. Devices not operating on Wi-Fi are often bridged over.

The operating frequencies are 2.4 GHz and 5 GHz, the former of which can see a lot of interference [43]. A lot of data can be sent over Wi-Fi and at fast rates, but this results in high power consumption [14, 45]. This is especially noticeable in mobile devices, but larger devices are bulky and costly.

Bluetooth.

Bluetooth is a short-ranged technology designed to eliminate cords, such as for connecting headsets and radios or allowing hands-free device access [45]. The main use of Bluetooth for home automation is mainly peripheral devices like speakers that connect to a bridge allowing voice-activated controls to Bluetooth devices due to its short range.

Bluetooth operates on the 2.4 GHz frequency band like Wi-Fi, but is designed to consume less power, thus limiting its range to only about ten meters [46]. Bluetooth devices are widely available and are typically capable of measuring and reporting the RSS themselves [27].

Bluetooth Low Energy (BLE) transmits smaller amounts of data than does regular Bluetooth and is preferred for its low energy consumption. This is attractive, but most related works using Bluetooth or BLE technology apply a passive device-based approach [13, 14, 26, 27].

Zigbee.

One of the major home automation protocols, Zigbee is built on the IEEE 802.15.4 standard and has low cost, short delay, and low power consumption [45]. Its technologies operate on at least one of the three frequency bands: 868 MHz, 915 MHz, or 2.4 GHz [47], but more commonly use the 2.4 GHz frequency. The line-of-sight transmission range of IEEE 802.15.4 is 70 meters, or about 10 meters with obstructions, making successful packet delivery in a typical indoor environment limited [1].

Z-Wave.

Z-Wave operates on the 908.4 MHz frequency [47] and has a range up to 30 m [45]. Its protocol requires complex routing strategies and a confusing array of different types of network master, slaves and other modules [46].

Power Line Technologies.

Power line-protocol technologies like X10 are part of an older home automation approach. Instead of using wireless radio waves, these technologies communicate using the building's electrical power line (PL). Typical power lines have a slow, limited speed, but installing high-speed alternatives have high installation costs [46]. Other than by monitoring human interaction with devices connected on the PL, PL technology potential in a DfP system is limited.

Insteon.

A dual-band home automation protocol, Insteon simultaneously operates on the 915 MHz RF band and on the PL, increasing its reliability [48]. It is a "responsive, easy-to-install, simple-to-operate, reliable, and affordable technology for home automation" with many documents available online [46].

Also, Insteon is capable of simulcasting, or the broadcasting of messages even when other devices are transmitting. This feature allows messages to have low probability of being cancelled [47]. This means that more messages can be broadcast in an environment. With the knowledge that Insteon's security is flawed [49], this indicates that Insteon technology can be exploitable for DfP.

Summary.

Lone sensors are too simple for exploitation, and non-radio technologies like cameras or the X10 have high processing costs, line-of-sight requirements, high installation costs, or are not as ideal as radio-based technologies.

Wi-Fi is popular, as is the 2.4 GHz frequency band on which it and other technologies like Bluetooth and Zigbee operate. However, many interference sources and overcrowding on this frequency make these technologies less attractive. Z-Wave and Insteon, on the other hand, operate on bands with 908 MHz and 915 MHz center frequencies, respectively. However, Z-Wave has a complicated routing protocol, whereas Insteon simulcasts its messages, which means more RF signals can be exploited in an Insteon network.

RSS DfP systems work because the human body attenuates radio signals. The 2.4 GHz band is often used for DfP because the human body contains about 70% water, and the resonance frequency of water is 2.4 GHz [8, 11, 18, 19]. However, the human body also attenuates other frequency signals, including the 915 MHz signal at approximately

Table 1. Comparison of some home automation technologies for DfP applications

Technology	Freq. Band	Pros	Cons
Sensors	Varies	- small - low unit cost	- limited use and capability
Cameras	Wi-Fi (typically)	- capable of detection and localization of stationary and moving targets	- privacy concerns - computational costs -requires LOS
Wi-Fi	802.11 2.4 GHz	- very common - can send large data	- interference - high power consumption
Bluetooth	802.15.4 2.4 GHz	- consumes low power - intrinsic RSS capability	- short range - mainly device-based approaches
Zigbee	802.15.4 2.4 GHz (typically)	- low cost - short delay - low power consumption	- only 10 m obstructed range
Z-Wave	908.4 MHz	- available documentation	- complex routing
X-10	Powerline		- slow - limited speed - high installation costs
Insteon	Dual-band 915 MHz (RF) 136 MHz (PL)	- affordable - simulcasting - available documentation	- no intrinsic RSS capability

0.3-0.5 dB per mm of tissue [50].

The 915 MHz band is one of the bands used by Zigbee technology, but Zigbee devices more commonly prefer the 2.4 GHz band. This and the greater availability of literature on Insteon technology makes Insteon more promising as a network to exploit.

2.5 Insteon Devices

Section 2.4 compared different home automation technologies and argued for Insteon because it does not operate on the 2.4 GHz frequency band that sees a lot of interference, is capable of simulcasting, and is a simple-concept technology with vast and readily-available documentation. In fact, the information is this section is largely from Insteon documents [51–53]. This section discusses the Insteon technology more

in-depth with a focus on Insteon communication protocols, which determine their capabilities. Familiarization with the protocols will help formulate an approach for deploying an RSS-based DfP system on the Insteon home automation network.

2.5.1 Intra-network Communication.

Insteon uses both PL and RF for communication [51]. As previously established, the focus of the research will be on the radio side, but understanding both protocols is important. The Insteon protocols for communicating between Insteon devices over PL or RF change slightly. Thus, only Insteon RF devices communicate with other Insteon RF devices, and only Insteon PL devices communicate with other Insteon PL devices.

Dual-band Insteon devices capable of both RF and PL communication use both protocols, but separate the two media. This is useful when PL signals originate on the opposite PL phase from the receiver, which would cause the signals would be strongly attenuated, but a single dual-band device installed on the PL phase can bridge the message to RF. If a message is relayed to another device, then two things can occur. If the message is received on the PL, then the device will relay the message on RF before relaying it on PL in the next time-slot [51]. If the message is received on RF, then it is relayed via PL in the next time-slot and then immediately relayed via RF. This allows re-synchronization at the earliest opportunity [51].

Power line communication describes the movement of electrons across a PL. In most North American homes, these power lines are typically two-wire 110 VAC lines connected at a main electrical junction box. On the physical layer, the Insteon PL protocol operates using the 131.65 KHz frequency band with a BPSK modulation [51]. PL messages are broken up into five or eleven packets [52]. Insteon claims that over the PL, Standard messages take 50 ms to send, while Extended messages take 108.33 ms.

Insteon RF physical layer communication operates at a 915 MHz center frequency

using FSK modulation [51]. The instantaneous bandwidth at this frequency is 3.84 GHz. The sensitivity of the communication is -103 dBm with a range of up to 150 feet of unobstructed line-of-sight. Over RF, messages are not broken up into packets due to faster messaging. Insteon claims that it takes 2.708 ms to send a Standard message and 5.625 ms to send an Extended message. Since zero crossings occur every 8.333 ms, either message can be sent during one PL half-cycle.

Despite these differences, both Insteon protocols have a common, general framework, including their message structure and their message repeating, signaling, and simulcasting features.

Message Structure.

Two types of Insteon messages exist: the 10-byte Standard message and the 24-byte Extended message [51]. The fields for both include a source address, a destination address, flags, command bytes, and a cyclic redundancy check (CRC) byte for message integrity. The Extended message additionally contains fourteen bytes for User Data, which is used for downloads, uploads, encryption, and other advanced applications. Both message fields are displayed in Table 2.

The source ('From') address is the device ID of the originator node. The destination ('To') address can be the destination node's device ID for a Direct message, the device type, sub-type, and firmware version for a Broadcast message, or the group number for a Group Broadcast message [52].

The 'Flags' field has a three-bit message type (the left-most bit is Broadcast or Not-Acknowledge (NAK), followed by the Group flag, the Acknowledge (ACK)) flag, a one-bit extended flag (0 for Standard messages), a two-bit 'Hops Left' field counted down on each transmission, and a two-bit 'Max Hops' field [52].

Broadcast messages contain general information with no specific destination and

Table 2. Insteon message fields [51]

Insteon Standard Message (10 Bytes)				
3 Bytes From Address	3 Bytes To Address	1 Byte Flags	2 Bytes Command 1, 2	1 Byte CRC

(a) Standard

Insteon Extended Message (24 Bytes)					
3 Bytes From Address	3 Bytes To Address	1 Byte Flags	2 Bytes Command 1, 2	14 Bytes User Data	1 Byte CRC

(b) Extended

Table 3. Possible flag bytes for Insteon messages [51]

Bytes	Description
0xC#	Group Broadcast
0x4#	Direct Group Cleanup
0x6#	Direct GC ACK
0xE#	Direct GC NAK
0x8#	Broadcast
0x0#	Direct
0x2#	Direct ACK
0xA#	Direct NAK

are used extensively during device linking. Group Broadcast messages are directed to linked devices in a group. Direct messages are point-to-point, and ACK responses are required for reliability. There are no ACKs or NAKs for Broadcast or Group Broadcast messages. However, after a Group Broadcast message is sent, the origin node sends a Direct 'Group Cleanup' message to each member of the group individually and waits for the individual acknowledgements [52]. All possible flag codes are given in Table 3. The octothorps ('#') indicate an unknown nibble from message re-transmission flags. Message re-transmission is described in more detail in this section.

The 'Command' field consists of two possible commands, but some functions only require one command. The second command may be ignored or may contain processed information. Some commands relevant to this research are listed in Table 4.

Table 4. Example Insteon command codes and message bytes [51]

Cmd1	Cmd2	Description
0x11	Group No. or On level	Light ON
0x12	Group No. or Ignored	Light ON Fast
0x13	Group No. or Ignored	Light OFF
0x14	Group No. or Ignored	Light OFF Fast
0x10	0x00	ID Request
0x19	Depends	Light Status Request

The CRC is first computed by the transmitting device. A receiving device computes its own CRC of the message and compares it to the one in the message field [52].

Message Repeating in Insteon Peer Networks.

Insteon devices act as peers able to send, receive, or relay messages [46]. They can repeat messages to *relay* the message to a node out of range or to *re-transmit* the message due to non-acknowledgement. This is particularly useful in situations where devices are out-of-range or their signals are too attenuated to receive messages from the originating device. This process may need more than one intermediate node and have more than one path, but these allow for higher probability that a message is successfully received at the destination node.

Re-transmissions occur only in direct messaging, when the origin device fails to receive an acknowledgement from the destination node about the original message [52]. In such cases, the origin node will resend the message up to five times. This is separate from repeating, so the node can send out even more messages.

Message relay, or hopping, occurs naturally in the simulcasting network. In order to prevent network saturation, only up to three hopping repeats are allowed [52].

The records of repeating messages are kept in the Max Hops and Hops Left fields of the Flag field (its bottom nibble) [52]. Message originators often set the two fields equal to each other. If Hops Left is non-zero, the device will repeat the message. When a mes-

sage is received with a Hops Left value of zero, the message is not relayed. Destination nodes do not relay messages designated to them.

Every time a message failed because it took more hops than the number in Max Hops, then this number increases up to the limit of three [52]. Therefore, a message always hops at least once, with at most four hops. After five retries, if the destination node's acknowledgement does not arrive at the origin, then this is noted. As a result, Insteon suggests that it is good practice to unlink devices from the network when physically removing them to avoid unnecessary message re-transmission.

PL-connected Insteon devices know when messages are not received due to being tied to a universal clock–the zero-line crossing [52]. A powerline cycle is 1/60 Hz. A time-slot is six or thirteen PL half-cycles for Standard or Extended messages, respectively. During a single time-slot, a message can be transmitted, re-transmitted, or acknowledged.

According to ref. [49], Insteon devices send out four different signal bursts. The difference is characterized by their duration (34, 84, 134, and 184 ms). It is not clear when or why these different signals are sent out.

Simulcasting and Signaling.

Data is transmitted with the most significant bit sent first. Wait times after sending PL messages allow RF devices to re-transmit a message. Data in Insteon messages do not route. Instead, Insteon devices simulcast with reference to a global clock–the PL zero crossing. Simulcasting is when Insteon devices send messages out even when other devices may be transmitting. As a result, concerns of cancellation are "practically impossible" according to Insteon, because they require two messages to be absolutely 180 degrees out of phase with the same amplitudes.

2.5.2 Internet Communication.

Software Application.

Few Insteon devices are designed to be Internet-enabled, but the Insteon Hub can connect a typical network of Insteon devices to the Internet. As its name implies, the Insteon Hub acts as an Insteon-to-Ethernet bridge, connecting a network of Insteon devices to the home Internet. This allows, for example, a user to use a Wi-Fi-connected app on their phone to turn the living room light on instead of manually pressing the button.

The app interfaces the user to the Hub, and allows any number of devices to connect to the Insteon network. Scheduling is available, alerts can be sent via push notifications or email alerts, and camera footage and control are available. But while most users use the app, the modem located in the Hub has a serial communication protocol with a specific command syntax.

The command syntax is a URL: `<ipaddress>/X?YYYYY=I=X`, where '`X`' is a number ('`0`' if short command, i.e. does not include destination and flags or '`3`' if a full command); '`YYYYY`' is the command in bytes; and the suffix is always '`=I=X`' [53]. The command is not limited to five characters as this example suggests. To better understand this, the Uniform Resource Identifiers (URIs) for some device control commands are presented in Table 5. Also, the URI command to view the Hub's buffer is `<ipaddress>/buffstatus.xml`, while the command to clear the buffer is `<ipaddress>/1?XB=M=1`.

2.5.3 Security and Exploitation Potential.

Network security for Insteon devices is claimed to be maintained through Linking Control and through Encryption within Extended Messages [51]. Linking Control is enforced by requiring physical possession of devices and masking non-linked network traffic for messages relayed outside the network. Insteon devices do not act on commands from unlinked devices. Linking devices requires two options: either the user physically

Table 5. Insteon serial communication URI examples for control commands [53]

Command	Cmd1	Cmd2	URI Example & Explanation
On	0x11	Group No. or On level	.../3?0262347cfd0F11FF=I=3 Increment brightness to On level 0xFF (highest) for device 34.7C.FD
Fast On	0x12	Group No. or Ignored	.../0?1205=I=0 Immediately turn on all devices in group 05
Off	0x13	Group No. or Ignored	.../3?0262347cfd0F1300=I=3 Turn device 34.7C.FD on to 0xAA brightness immediately
Fast Off	0x14	Group No. or Ignored	.../0?1305=I=0 Instantly decrease brightness to Off for all devices in group 05
Start Dim/Brt	0x17	01 = bright 00 = dim	.../3?0262347cfd0F1701=I=3 Enter brightening phase for device 34.7C.FD
Stop Dim/Brt	0x18	Ignored	.../0?1802=I=0 Cancel phase
ID Request	0x10	0x00	.../3?026240506D0F1000=I=3 Request device 40.50.6D to send device identification details
Status Request	0x19	Depends	.../3?02623966990F1900=I=3 Request device 39.66.99 to send On-level of light in Cmd2

presses a button on the device or the user must read the device address from the label on the device and manually enter it when prompted. Additionally, Insteon devices can connect outside the Insteon network with bridges, and security is maintained by masking the two high-bytes in the address fields of messages unless the traffic is from an Insteon device already linked or from a device that already knows the address.

Exploiting a home automation network is important because a home automation network is pre-existing and functions regularly, with signals already being communicated across the network ready to be captured for RSS readings. With no need to install a DfP-specific network, the elimination of costs is a major factor.

Unfortunately, Insteon devices do not collect or transmit messages containing the signal strengths of their received signals. A crude indicator of RSS could be monitoring dropped packets, but this would be difficult to exploit given Insteon's simulcasting and message relaying functions. Therefore, a cheap alternative should be sought for the purposes of this research. Note that other than the network security details for the Insteon network presented above, no Insteon documentation exists concerning security, especially for Insteon Hubs and the associated software applications.

2.6 Summary

Indoor, radio-based, DfP techniques are low-cost, perform covertly, and can be installed on a WSN to detect changes in the environment and approximate the target location passively, without requiring any devices to be attached.

Measuring RSS is preferred for DfP techniques because the measurement is easy to measure and the signals already exist in wireless home automation networks, which lowers costs. For one-receiver, RSS-based DfP detection, the baseline signal comparison approach is a simple approach. For RSS-based, multiple-receiver DfP localization, the probabilistic fingerprinting approach outperforms the deterministic approach and can be used for networks with fewer nodes, as is the case in home automation networks. Since this research plans to exploit a home automation network's nodes for DfP detection and localization, the centralized approach is attractive because it allows all receiving nodes to share information, thereby increasing accuracy.

Insteon technology is chosen among the home automation technologies because of its simulcasting feature and the availability of papers detailing its technology. However, there is the concern that Insteon devices are not capable of recording RSS. A cheap alternative should be sought for the purposes of this research. Also noted is that no Insteon documentation exists concerning security of Insteon Hubs and the associated software applications.

III. Methodology

The goal of the research is to provide a proof of concept for achieving DfP detection and localization through the exploitation of a home automation network's RF signals. At minimum, the proof of concept involves demonstrating that the attenuation from human presence of a selected home automation network's signals can be used in detection and localization. Successful detection and accurate localization of a human target using a network of these nodes are also sought.

Three experiments are planned to establish this proof of concept. First, successful detection of human presence in a room with a simple one-transmitter, one-receiver setup is planned. Second, accurate cell-based localization of a human target in a multiple receiver setup using fingerprinting techniques is planned. Finally, the localization system is moved to other rooms to test for robustness despite varying room setup.

Before these experiments can be carried out, preliminary work must be conducted. Exploitation of the Insteon network for this research physically involves the monitoring of the Insteon RF messages for RSS, but Section 2.5 pointed out that Insteon devices do not collect or transmit RSS, so an alternative must be developed. The methods for detection and localization must also be developed before deployment.

Therefore, the approach for establishing a proof of concept is to first undergo preliminary work establishing the methods for exploiting the Insteon home automation network and for conducting the detection and localization. Afterward, the three primary experiments will be conducted to demonstrate success and accuracy of an RSS-based DfP system that exploits a home automation network's RF communications. This chapter details the assumptions, equipment, experimental approach, evaluation metrics, and environments used in developing and verifying the proof of concept.

3.1 Assumptions

The assumptions below apply to the entire proof of concept system in order to limit the number of factors and variables to consider and to focus instead on just establishing a general proof of concept.

Assumption 1 *Height is insignificant and all components are at comparable height.*

Assumption 2 *There is negligible variability in how human targets of different sizes attenuate signals.*

Assumption 3 *The network to be exploited has enough transmitters to send signals across the entire area of interest.*

Assumption 4 *The home automation devices are able to transmit and receive in any direction. Receiving nodes are also able to receive in any direction.*

Assumption 5 *Only one person can be in the area to be detected or localized at a time.*

Assumption 6 *Localization of a target within a cell of space provides sufficient precision.*

3.2 Equipment

The proof of concept is a system of nodes capable of detecting signal strength and of sending this information to a database for processing. The system should have affordable hardware, have low installation costs, and be unobtrusive and inconspicuous. The processing of data should happen quickly with alerts to detection and localization visibly printed somewhere. This section presents all the devices and related materials used in the research to form the RSS DfP detection and localization system for the proof of concept.

3.2.1 Insteon Home Automation Network.

A home automation network is needed because its RF signals are the focus of exploitation by the system. As discussed in Section 2.4, the Insteon network is used in this research mainly because it transmits on the less-frequently-used 915 MHz band. Let an Insteon network be comprised of an Insteon Hub and any number of Insteon RF devices.

The Insteon Hub (model no. 2245-222) is a central command device for Insteon networks. As discussed in Section 2.5.2, the Hub connects the Insteon network to the Internet. It is also a dual-band device. All Insteon devices have their own ID, and that of the Hub used in this research is `39.21.82`. The Windows Store version and version 1.9.8 of the Android software application, "INSTEON for Hub," were used in preliminary work and for network setup.

Most other Insteon devices are just dual-band devices with the same basic capabilities. Without loss of generality, the Insteon Dimmer Keypad 6-Button (model no. 2334-232) acts as other devices in the Insteon network. This is a popular smart device often installed in place of a standard wall switch and which can link other devices in the network.

The Keypads used in this research are modified for portability; they are installed in a PVC outlet box with a two-gang wall plate (see Fig. 5). The Keypad sits in one gang and a duplex outlet sits in the other. Power is supplied by a six-foot outdoor power cord that connects to a typical outlet. The unique IDs of all six Keypads used in this research are listed in Table 6.

For the purposes of this research, all Insteon devices are considered transmitters. The Hubs are labeled in diagrams by an 'H' or 'TX'. Keypads are also positioned with receiving nodes to simulate a future development where these devices have DfP capabilities. For purposes of this research, the assignment of Keypads to a general number and the assignment of Keypads to groups for Group messaging are shown in Table 6.

(a) Hub (2245-222)

(b) Dimmer Keypad 6-Button (2334-232)

Figure 4. Insteon devices used to comprise an Internet-enabled home automation network

Figure 5. Insteon Dimmer Keypad encased in PVC outlet box with additional outlets

Table 6. Keypad IDs and assignments ('x' is 'belongs to') for Group messaging

		Group No. (Firmware-designated No.)				
No.	Device ID	2 (05)	3 (06)	4 (07)	5 (10)	6 (11)
1	34.7C.FD	x	x	x	x	x
2	34.E0.2A	x	x	x	x	x
3	40.4D.DF		x	x	x	x
4	40.50.6D			x	x	x
5	39.66.99				x	x
6	39.45.65					x

3.2.2 Data Collection and Processing System.

Any data collection and processing system (DCPS) should be comprised of RSS receivers, a location to store the database, and a processor for applying the detection and localization methods. Ideally, the DCPS utilizes the same resources as the home automation network devices. However, Section 2.5.1 discussed how Insteon messages do not have a field for carrying RSS indicators (RSSI) that can be used for this research. The proof of concept only seeks to establish that exploitation of the home automation network is possible for detection and localization in hopes of future joining of the system with existing home automation devices. Thus, this proof of concept must have an alternate method to detect RSS.

This alternative could be RSS receivers capable of collecting 915 MHz signals and of connecting to a processing database. Software-defined radios (SDRs) are such devices. As a result, the nodes used in this research are NooElec NESDR Mini 2 SDR USB receivers, which are light-weight, low-cost, modified USB dongles with SDR capabilities [54]. The nominal frequency range of the receiver is approximately 25-1750 MHz, which includes the Insteon RF frequency. The instantaneous bandwidth is 2.4 GHz (nominal) with a maximum of 3.2 MHz (Insteon has a 3.8 GHz instantaneous RF bandwidth). These receivers also have MATLAB® support, allowing for real-time wireless signal processing. The devices can also be equipped with a telescopic monopole antenna with a meter-long cord (Figure 6).

Because they are USB devices, the receivers connect easily to a computer that can act as the database and processing unit. For simplicity, a laptop is used. To allow more receivers to connect to a single laptop, a USB hub is connected to the laptop. USB extenders are also used to position the receivers across the room.

However, it should be noted that in a truly deployed scenario, the receivers would be the actual home automation devices themselves and thus not need to be extended or

Figure 6. NooElec receiver and antenna used for data collection

connected through wires. As a result, the devices would be able to transmit their results wirelessly (or in Insteon's case, either through the PL or over RF) to the database and processing unit. The processing unit (e.g. the Hub) would then collect these signals and perform the necessary algorithms for detection or localization.

Therefore, the DCPS for this proof of concept consists of NooElec NESDR USB receivers and a laptop. The DCPS may be referred to as the "receiving unit" or the "receiving system." The NooElec receivers are referred to as "receivers" or "nodes", which are not to be confused with the Insteon devices. In diagrams, a node is labeled as 'RX'.

3.2.3 Miscellaneous.

In lieu of actual human targets, foam blocks are used for simplicity. Specifically, ECCOSORB® AN-79 blocks (Figure 7) are used as an alternative to human targets. Each foam block is a 2 ft x 2 ft (0.61 m x 0.61 m) square block of 4.49-inch (11.4-cm) width. They consist of carbon-treated polyurethane foam treated and absorb microwaves ranging from 600 MHz - 40 GHz [55]. The number of foam blocks estimated to simulate human attenuation is one. Two blocks will be vertically stacked to simulate a human body. The terms "target" and "human" in this research are interchangeable with the foam blocks.

Figure 7. Section of ECCOSORB® AN-79 foam block

3.3 Methods and Evaluation Metrics

Two main types of experiments are conducted in Chapter V: detection with a simple setup and localization with multiple nodes. The localization experiment is divided into two sub-experiments, which are introduced in the next section. This section presents the experimental approach and evaluation metrics for detection and localization using the proof of concept's RSS-based DfP system.

3.3.1 Detection with a Simple Setup.

Concept and Approach.

The minimal approach to establishing the proof of concept is to perform detection using a tripwire configuration. This consists of only one transmitting device and one receiving node with no obstructions between them. When the devices are communicating, RF signals exist. If a human subject stands between these two devices, then their presence should disrupt the RF field by absorbing and/or reflecting the signal. As a result, the signal strength received at the receiving node should be less than when there was no target between the devices.

This attenuated signal is the basis for the tripwire detection experiments conducted in Section 5.1. If the baseline signal for when there is no target is known in addition to knowing the event signal for when there is a target, then a threshold can be used to perform detection. During detection, if the collected RSS falls inside this threshold, then a detection event has occurred. This method, the baseline threshold comparison, was introduced in Section 2.2.2.

The experiment for simple-setup detection is split into two cases: detection at the midpoint and detection along the LOS. The first case assumes that the human only appears in the middle of the setup. This is a prefatory case that does not require the system to factor in the target's distance from either device or from the LOS. If successful, this experiment can establish a single signal attenuation value and show that the basic detection model functions.

The second case expands on this and takes into consideration how the signal attenuation changes as the target gets closer to the source or the receiver while remaining on the LOS of the two devices. It is not expected that detection using this method is successful all along the LOS; e.g, when the target is close to either device, the signal may be too attenuated by the target to reach the receiving node. Therefore, this experiment can demonstrate the accuracy of the tripwire detection model and also investigate the effect of distance along the LOS on signal attenuation.

Further cases, including that of detection outside the LOS, are a problem better answered with localization using multiple receivers (see Section 3.3.2).

Variables and Equations.

For detection in general, the response variable is the mean RSS, $\overline{R}$, or the single RSS value, R, both of which are measured in dB. R is calculated using the following equation:

$$R = 20\log_{10}\left|\rho\right| \tag{11}$$

where ρ is the relative amplitude of the signal. In the case of midpoint detection, the control variable is target presence, which is a Boolean variable denoted by subscripts b for no presence (baseline) and e for presence (event). The mean RSS for when there is human presence, $\overline{R}_b$, and for when there is not, $\overline{R}_e$, are compared for signal attenuation and threshold calibration (see Definition 1 below).

LOS detection has two control variables: human presence and the distance from the Hub. This distance, d, is a discrete variable belonging to a set of distances predetermined in the environment setup in Section 3.4.1. Mean RSS comparisons for signal attenuation levels and threshold calibrations occur at each d. Multipath and other effects are assumed to be negligible in the detection experiments.

Definition 1 *In a particular setup, the mean signal attenuation, $\overline{R}_a(d)$, of a target at a distance, d, along the LOS is the difference between the mean RSS for when there is no target present, $\overline{R}_b$, and that for when the target is present, $\overline{R}_e(d)$.*

$$\overline{R}_a(d) = \overline{R}_b - \overline{R}_e(d) \tag{12}$$

Both $\overline{R}_b$ and $\overline{R}_e$ are taken during calibration and provide the basis for the detection threshold. Assuming that the signal attenuation is statistically significant (i.e. not statistically equal to zero), a range for the threshold, $\delta r(d)$, can be constructed. The upper threshold separates baseline RSS from event RSS to determine detection, and the lower threshold is used to reduce false detection from outlier RSS not due to attenuation.

Definition 2 *The range of the detection threshold, $\delta r(d) = r^+(d)$ - $r^-(d)$, for distance d is determined using the following definitions for the upper and lower thresholds:*

$$r^+(d) = \min\left(\overline{R}_b - s_b\sigma_b(d), \overline{R}_e(d) + s_e\sigma_e(d)\right) \tag{13}$$

$$r^-(d) = \overline{R}_e(d) - s_e\sigma_e(d) \tag{14}$$

where $\sigma_b(d)$ and $\sigma_e(d)$ are the standard deviations of the RSS values, and s_b and s_e are a positive real number calibrated experimentally.

Then, during observation, the event of detection is defined as follows:

Definition 3 *The event that a target is detected occurs when the measured $R(d)$ is within the calibrated threshold for that d; i.e., at d, the detection event occurs when*

$$r^-(d) < R(d) < r^+(d) \tag{15}$$

For example, if at $d = 2$, the system calibrated $\overline{R}_b(2) \pm \sigma_b(2) = 50 \pm 2$ dB and $\overline{R}_e(2) \pm \sigma_e(2) = 45 \pm 1$ dB, the $\overline{R}_a(2) = 5$ dB. Also, if $s_b = 1$ and $s_e = 1$, the threshold would be $(r^-(2), r^+(2)) = (44, 46)$ dB. An RSS of $R(2) = 47$ dB would not be a detection event, but $R(2) = 45$ dB would. But if $s_b = 3$ and $s_e = 2$, the upper threshold would be $r^+(2) = 44$ dB. The method is also depicted in Figure 8.

Evaluation Metrics.

In the detection experiment, several results appear and are evaluated. One of these is $\overline{R}_a$. To compare the event signal against the baseline signal and determine that the signal attenuation calculated using (12) is significant enough to perform detection, a one-tailed, paired t-test is used. Several paired collections of $\overline{R}_b$ and $\overline{R}_e$ are collected for this.

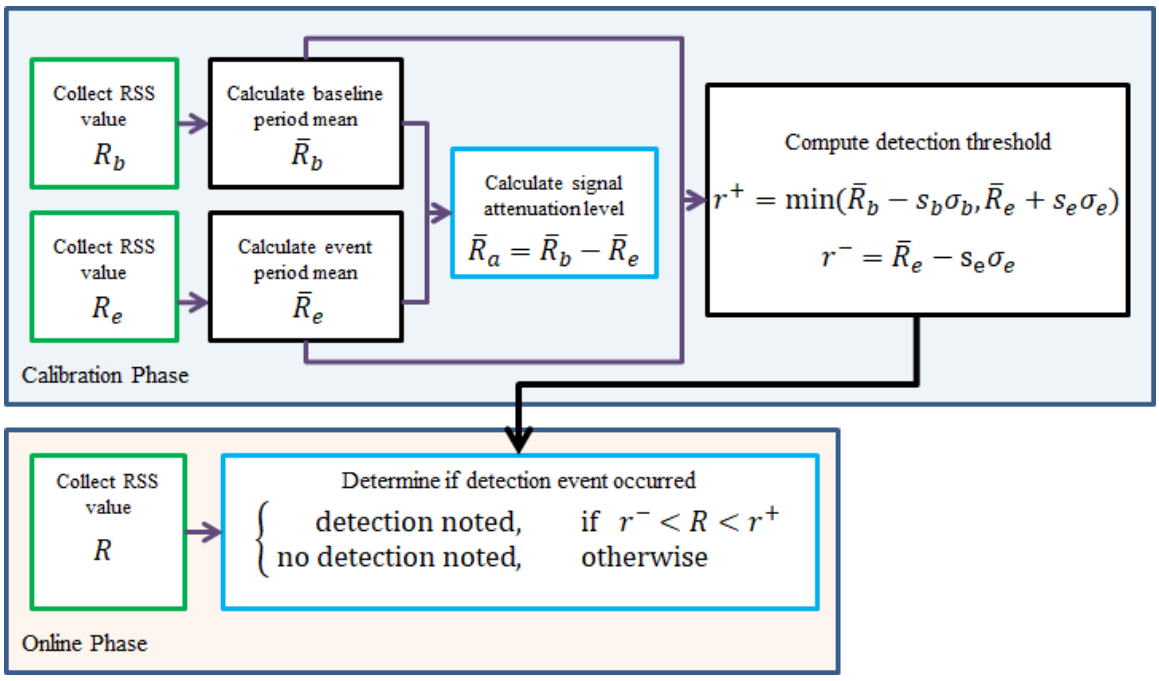


Figure 8. Flow chart depicting the detection method

Metric 1 *The significance of the signal attenuation level is determined using a one-tailed, paired t-test. Let* $\overline{X}_D(d) = \overline{R}_a(d) = \overline{R}_b - \overline{R}_e(d)$ *for some distance,* d*. The test hypotheses are:*

$$H_0 : \overline{X}_D = 0$$

$$H_A : \overline{X}_D > 0$$

This t-test is performed in MATLAB® *using the function*

```
ttest(R_o,R_e,'tail','right')
```

where `R_o` *is the vector containing all* $\overline{R}_b$ *and* `R_e` *is the vector containing all* $\overline{R}_e(d)$*.*

In cases where detection can be performed, the null hypothesis should be rejected at $\alpha = 0.05$; i.e., $\overline{R}_b$ is significantly larger than $\overline{R}_e(d)$, and their difference is a significant signal attenuation level. All $\overline{R}_a(d)$ will be presented in a table and an error bar chart. If these levels are significant, then detection thresholds can be calculated. The thresholds

for each d will be presented side-by-side in an error bar chart. This will help visualize the calibrated thresholds for comparison and to qualitatively predict performance of detection.

If paired calibrations are not necessary, then the one-tailed, unpaired t-test can be used to verify that $\overline{R}_a(d)$ is statistically significant.

Metric 2 *For unpaired calibrations, the signal attenuation at distance d is significant if the p-value is small in an unpaired t-test. Let $\overline{X}_1 = \overline{R}_b$ and $\overline{X}_2 = \overline{R}_e(d)$. Assuming that the true variances of $\overline{R}_b$ and $\overline{R}_e(d)$ are equal, then the test is conducted as follows:*

$$H_0 : \overline{X}_1 - \overline{X}_2 = 0$$

$$H_A : \overline{X}_1 - \overline{X}_2 > 0$$

$$\mathrm{df} = n_1 + n_2 - 2 \tag{16}$$

$$s_p = \sqrt{\frac{(n_1 - 1)s_{X_1}^2 + (n_2 - 1)s_{X_2}^2}{\mathrm{df}}} \tag{17}$$

$$t = \frac{\overline{X}_1 - \overline{X}_2}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{18}$$

After the t-statistic is computed, the following MATLAB® *command can be used to calculate the p-value:*

```
1-tcdf(abs(t),df))
```

The null hypothesis should again be rejected based on $\alpha = 0.05$.

In addition to these calibration results, the other results appear when actually conducting the detection experiment. Two evaluation metrics are used for evaluating the detection system: the probability of detection, P_D, and the probability of false positives, P_F. The goal is to have the probability of detection as close to 100% as possible and the probability of false positives as close to 0% as possible.

Metric 3 *The probability of detection, P_D is the likelihood that the system correctly detects a target.*

$$P_D = P(\text{Detection Noted} \mid \text{Target Present}) \tag{19}$$
$$= \frac{\#\text{ of detection noted when the target is present}}{\#\text{ of total trials when target is present}}$$

Metric 4 *The probability of false positives, P_F, is the probability that the system incorrectly identifies a measurement as an event even though there was no target present.*

$$P_F = P(\text{Detection Noted} \mid \text{Target Not Present}) \tag{20}$$
$$= \frac{\#\text{ of no detection noted when the target is not present}}{\#\text{ of trials when target is not present}}$$

Both of these metrics will be presented in a table for comparison of different d. Each distance is tested independently. The threshold parameters s_b and s_e are determined based on P_D and P_F, and a ROC graph will present the findings of this investigation.

3.3.2 Localization with Multiple Receivers.

Basic Concept and Approach.

Localization is a more practical expansion of detection in that it involves a larger area and approximates a target's location rather than just notes presence. Instead of one-dimensional LOS detection, localization is a two-dimensional problem. As a result, this problem benefits from multiple receiver nodes stationed around the area of interest. Each node in the area of interest collects its own RSS and computes the mean RSS over a period of time. Since the nodes are in different locations, each one estimates the signal strength in the room differently. For different target locations, these different values from each node can be stored in a database, which can then be used to match an

observed RSS to select the probable location of a target. This is known as fingerprinting, a topic introduced in Section 2.2.3.

Fingerprinting is an expansion of the baseline comparison that first calibrates a map containing unique signatures for various points from various nodes and then compares the raw data to classify the location. With home automation, the areas of interest are not pin-point locations, so the cell-based method discussed in Section 2.2.3 is an adequate approach.

In the offline phase, the area of interest is divided into cells. The location of both home automation devices and receiving nodes are not used in this method, but in this research, the receiving nodes are set on the outer edges of the area of interest (in practice, devices should be deployed in inconspicuous locations without impeding movement). Additionally, the nodes should be spread out and not clumped together (home automation devices are typically positioned in such a manner). A human target will be positioned in each of the cells to be calibrated, and data will be collected from the nodes to create RSS signatures at each cell. This is the offline data map. In the online phase, the target is standing at a single location and the observed data from all nodes is collected. To locate the target, this data is matched to the offline data to find the location of maximum probability.

Variables and Definitions.

The response variable in all localization experiments is also $\overline{R}$ or R. These are calculated the same way as in (11). The main control variables for all experiments is target location. The experiment in Section 5.2 additionally has target orientation as a control variable to evaluate the effect of target orientation on calibration and localization. The remaining experiments also vary the room setup as a control variable to test the robustness of the proof of concept.

If the calibration time for detection is determined to produce statistically similar mean RSS to those of larger periods, then the same can be said for the localization calibration time. As a result, this is not re-evaluated. Additionally, the signal attenuation levels are not of importance since thresholds are not used for localization. Instead, localization begins with the collection of RSS data for each possible target location from each node. Let there be C cells of interest and N receiving nodes deployed.

Let $\overline{\overline{R}}_e$ be the $(C+1)\times N$ offline data matrix of calibrated mean RSS for target presence in all $C+1$ cells (including the case where the target is not present in any cell) from all N nodes. Let $\overline{R}_e(c)$ be a length-N vector in $\overline{\overline{R}}_e$ containing the mean RSS values for target presence in cell c from all N nodes. Let $\overline{R}_e(c,n)$ be the single mean RSS value of the attenuated signal of a target standing in cell c taken from node n.

Offline calibration is performed as follows: for some time period, T, calibration data is taken from node n of a target standing in cell c. This value is averaged into $\overline{R}_e(c,n)$. This calibration is also performed for when the target is not in the area of interest as $\overline{R}_e(0,n)$. Each possible target location then has N mean RSS values, one from each node, and this is stored in $\overline{R}_e(c)$. When all data is collected for all nodes and all cells, then this data is stored in the matrix $\overline{\overline{R}}_e$. (See the left-hand side of Figure 9.)

In the online phase, all N nodes simultaneously collect RSS data starting at time t_0. Each node, n, computes and stores the mean RSS over a time period, $\tau < T$, in $\overline{R}(t_1,n)$ where time $t_1 = t_0 + m\cdot\tau$ for some integer m. The accumulation of $\overline{R}(t_1,n)$ for all nodes is $\overline{R}(t_1)$, a vector of size N. Then at some time t_1 during which human presence occurs, a cell in which human presence exists is selected by comparing the observed data vector $\overline{R}(t_1)$ to each of the $\overline{R}_e(c)$ vectors in $\overline{\overline{R}}_e$ and finding that cell at which the maximum probability of match occurs. (See the right-hand side of Figure 9)

The probability of match for a sample at some t_1 is defined by the average P-values calculated using two-sided p-tests comparing $\overline{R}(t_1,n)$ and $\overline{R}_e(c,n)$ for each node.

Definition 4 *To select the most probable target location, c_p, for the true location, c_{true}, the two-sided Welch's t-test is used. Since it is unknown whether or not the variance of the RSS for c_{true} is equal to that for c_p, the variances are estimated separately. Let $\sigma_e(c,n)$ and $n_e(c,n)$ be the standard deviation and number of steps from the offline collection and let $\sigma(t_1,n)$ and $n_o(t_1,n)$ be similar variables from the online collection for some n, c, and t_1. Then the two-sided Welch's t-test hypotheses are:*

$$H_0 : \overline{R}(t_1,n) = \overline{R}_e(c,n)$$

$$H_A : \overline{R}(t_1,n) \neq \overline{R}_e(c,n)$$

Let $s_1 = \sigma_e(t_1,n)$, $s_2 = \sigma(c,n)$, $n_1 = n_o(t_1,n)$, and $n_2 = n_e(c,n)$. Then the t-statistic is calculated as follows:

$$t = \frac{\overline{R}_e(c,n) - \overline{R}(t_1,n)}{s_{\overline{\Delta}}} \tag{21}$$

where

$$s_{\overline{\Delta}} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} \tag{22}$$

and the degrees of freedom used in significance testing are

$$\mathrm{df} = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)}{\frac{(s_1^2/n_1)^2}{n_1-1} + \frac{(s_2^2/n_2)^2}{n_2-1}} \tag{23}$$

Assuming that H_0 is true, the Welch's t-test calculates a p-value that represents the probability that the observed value is close to the offline value. The MATLAB® *function for the t-distribution is again used to find the p-value:*

```
2 * (1 - tcdf(abs(t), df))
```

Definition 5 *Let $P(c,n)$ be the probability that the observed value is close to the offline value. Then the selected location cell, c_p, at t_1 is some c for which the probability of match for a sample at some time t_1 is the maximum; i.e., $\exists$ c such that*

$$c_p = \underset{c \in \{0,1,...,C\}}{\arg\max} \left(\frac{1}{N} \sum_{n=1}^{N} P(c,n) \right) \tag{24}$$

But if $c_p = 0$, then no presence was detected.

Figure 9 depicts the localization method.

Evaluation Metrics.

Accuracy, P_A is the primary metric for evaluating the localization system. It determines how many correct selections the system makes. The goal for this value should be 100%, but at minimum, the system accuracy should at least be greater than the probability of random guessing $(1/(C+1))$.

Metric 5 *The probability of correct localization is the likelihood that the system correctly selects the true target cell in all $c \in \{0,1,...,C\}$ (i.e. including the empty case):*

$$P_A = \frac{\text{\# of total correct cell selections}}{\text{\# of total trials}} \tag{25}$$

In addition to accuracy, the system is evaluated by the overall true positive rate and overall false positive rate. The goal of the true positive rate is 100% and of the false positive rate is 0%. These rates are denoted as P_D and P_F, respectively, and are similar to the metrics for detection, but are defined with more conditions.

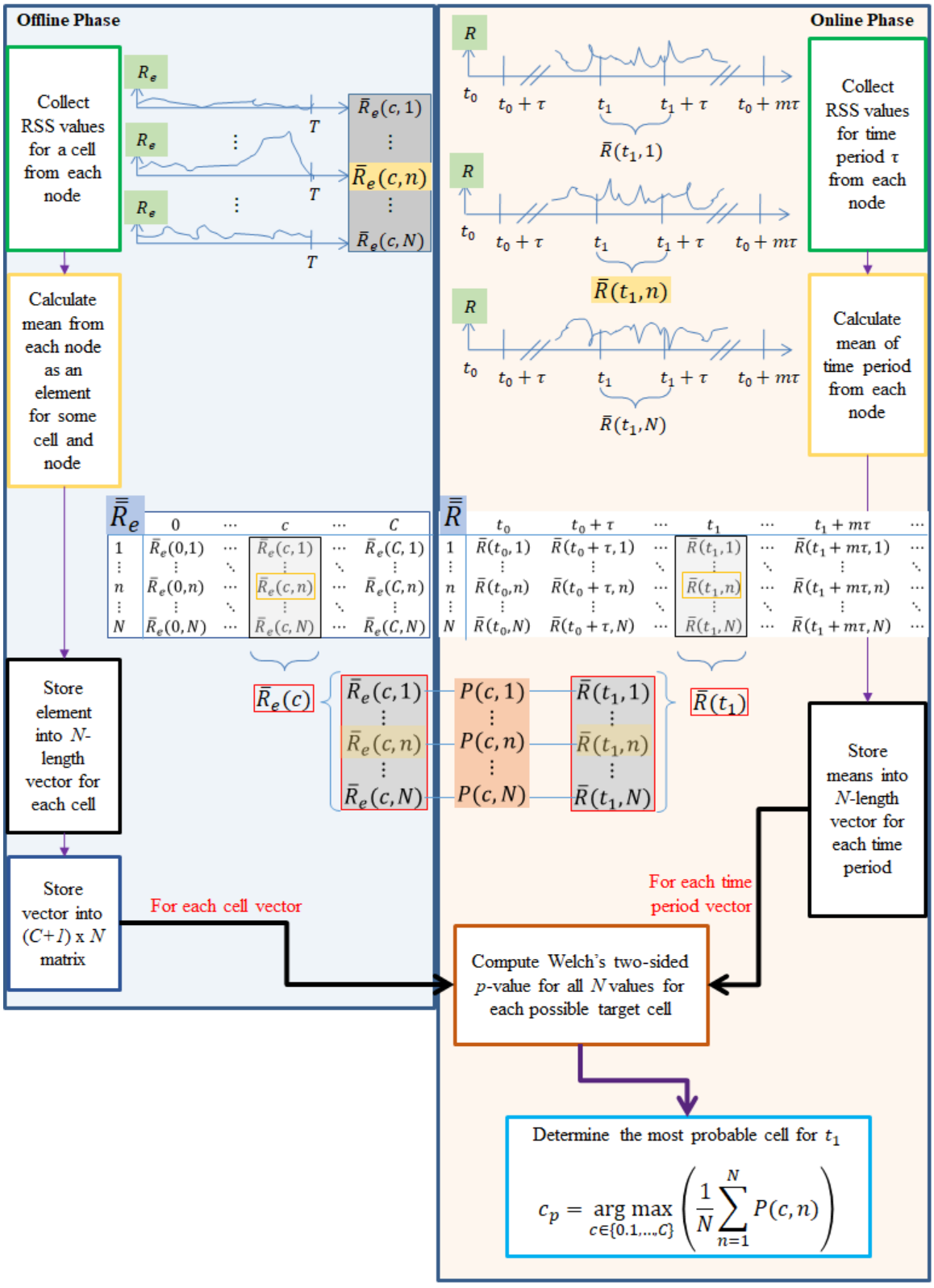


Figure 9. Flow chart depicting the localization method

Metric 6 *Let D be the event that the localization is successful when the target is in a cell $c \in \{1,2,...,C\}$. Then the overall true positive rate is the probability of D.*

$$P_D = P(c_p = c_{true} \in \{1,2,...,C\} \mid c_{true} \in \{1,2,...,C\}) \tag{26}$$
$$= \frac{\text{\# of correct cell selections when } c_{true} \neq 0}{\text{\# of trials when } c_{true} \neq 0}$$

Metric 7 *Let F be the event that localization is incorrect when the target is in the empty case cell, $c = 0$. Then the overall false positive rate is the probability of F.*

$$P_F = P(c_p \in \{1,2,...,C\} \mid c_{true} = 0) \tag{27}$$
$$= \frac{\text{\# of incorrect cell selections when } c_{true} = 0}{\text{\# of trials when } c_{true} = 0}$$

For a more detailed evaluation of localization, the true positive and false positive rates are also measured for each possible cell. Each possible location, c, has its own $P_D(c)$ and a $P_F(c)$ with similar goals as their overall counterparts above.

Metric 8 *The true positive rate for a possible location, c, is the probability that the most probable selected cell, c_p, is the true target location, c_{true}:*

$$P_D(c) = P(c_p = c \mid c_{true} = c) \tag{28}$$
$$= \frac{\text{\# of times that } c_p = c \text{ when } c_{true} = c}{\text{\# of trials when } c_{true} = c}$$

Metric 9 *The false positive rate for a possible location, c, is the probability that the most probable selected cell, c_p, is not the true target location, c_{true}:*

$$P_F(c) = P(c_p = c \mid c_{true} \neq c) \tag{29}$$
$$= \frac{\text{\# of times that } c_p = c \text{ when } c_{true} \neq c}{\text{\# of trials when } c_{true} \neq c}$$

3.4 Environments and Scenarios

This section presents the experimental setups of the two main experiments conducted in Chapter V. The simple setup for detection used in the experiments in Section 5.1 is discussed first. The second main experiment is multi-node localization. Setup 2 is used in the first localization sub-experiment in Section 5.2. Setup 3 is used in the remaining sub-experiments in Section 5.4 and has three cases of varying room types.

3.4.1 Setup 1: Simple Setup in Empty Apartment Room.

For simple detection, the equipment is just one Hub and one receiver. Since the experiment that uses this scenario only attempts to demonstrate proof-of-concept success of the simplest case, an empty apartment room is used. It is a 10 ft x 10 ft (3.0 m x 3.0 m) room with an additional 2.5 ft x 4.3 ft (0.76 m x 1.3 m) section (see Figure 10). The walls are plaster and the floor is carpeted. There are two large, two-pane windows along one wall and two doors. This room has no furniture or fixtures other than the ceiling fan; the room is in the exact condition as when a new inhabitant first enters the unfurnished room. This minimizes multipath effects and reflections that would occur with obstacles.

The devices in this scenario are set up as in Figure 10. Because this is the tripwire case, the node is set up across from the transmitter. The center of the room with the devices at the midpoint of either wall edge is selected, but the method should function in any case. For the experiments in Section 5.1, no other devices should be in the room in order to maintain the effort of minimizing other effects on the signal, like reflection from objects. Therefore, the processing unit is located outside the closed door.

The set of distances tested is $D_H = \{1.0, 2.5, 4.0, 5.0, 6.0, 7.5, 9.0\}$ ft ($D_H = \{0.30, 0.76, 1.22, 1.52, 1.83, 2.29, 2.74\}$ m) and is noted in Figure 10. Midpoint detection is conducted at 5 ft (1.5 m). LOS detection is conducted at all $d \in D_H$. The target should symmetrically and perpendicularly straddle the LOS to block the direct signal as much as possible.

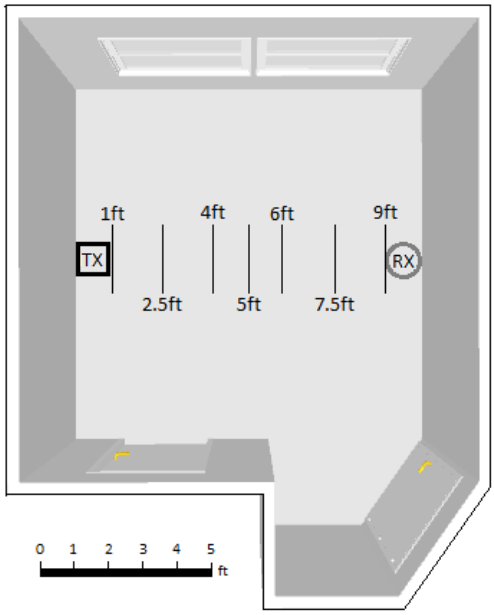


Figure 10. Diagram of apartment room without furniture, with one transmitting device (TX) and one receiving device (RX), and with LOS distances marked

3.4.2 Setup 2: Multi-Node Setups.

To establish that localization is also possible with the proof of concept, the same empty apartment room is used. Since this is a more practical scenario than the tripwire scenario, a typical home automation network is set up. Assume that a typical Insteon network for a single apartment room of this size is one Hub and three Keypads. The home automation devices are set up typically where a smart light switch or smart outlet would exist and are spread out across the room. Since the DfP system would ideally be deployed as a home automation device, the receivers are positioned next to the Keypads and their units are labeled as 'N#'. The setup is seen in Figure 11.

Note that the area of interest in the room is only the 10 ft x 10 ft (3.0 m x 3.0 m) square area. Since multipath effects are not expected to be minimized in localization, the door is not closed, and the DCPS was located inside the room but outside the area of interest.

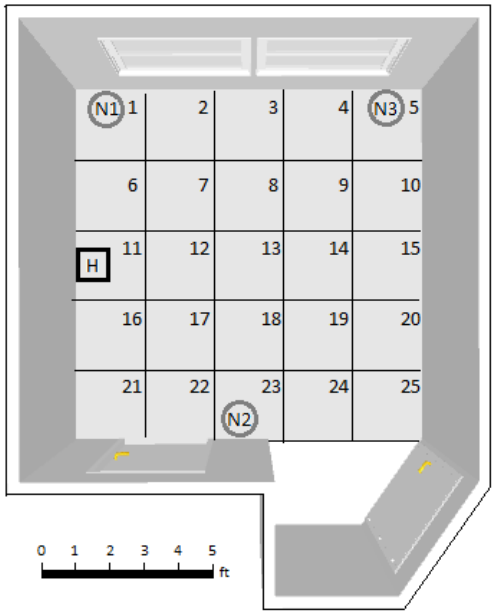


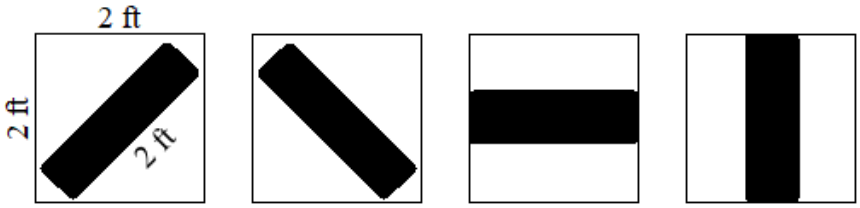
Figure 11. Diagram of Room A including cell division and node setup (H is an Insteon Hub (transmitter), N# are receiver nodes)

The division of the room was chosen to be 2 ft x 2 ft (0.61 x 0.61 m) cells. These dimensions for the cell were chosen to be the same length as the foam block. This resulted in a total of $C = 25$ cells, which are labeled in Figure 11.

Additionally, in the first part of the section, the orientation of the foam block wall will vary as a factor to examine. Four orientations are used and are depicted in Figure 12 with respect to the orientation of the room diagrams. These target orientations are referred to as 'O#' in the experiments.

3.4.3 Setup 3: Multi-Node Setups in Various Rooms.

To determine the robustness of the proof of concept, the system must be deployed in rooms with other properties. Three rooms are used in the experiments in Section 5.4. The number of devices and nodes should not vary in these rooms since they all have the same number of outlets and light switches. The major difference between these rooms



(a) O1 (b) O2 (c) O3 (d) O4

Figure 12. Possible foam block target orientations in a cell with respect to room layout

and Room A is the existence of obstacles. Obstacles can absorb and reflect signals to varying degrees, thus either exacerbating the results or improving them.

Room A: Empty Apartment Room.

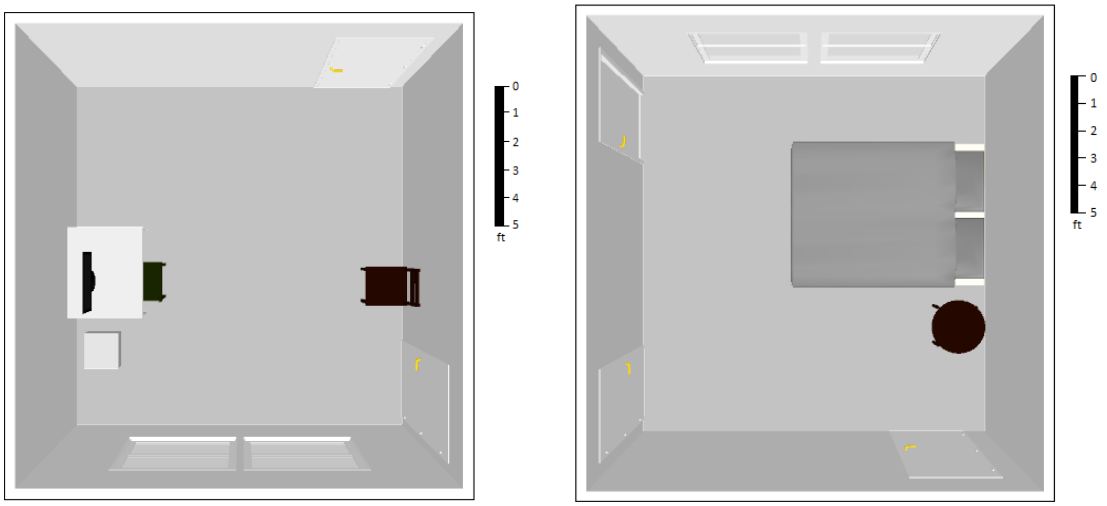
Room A is exactly the same as the furniture-less, three-node setup in Scenario 2 (see Figure 10) and serves as a baseline comparison.

Room B: Apartment Room with Few Obstructions.

Room B is similar in construction materials as Room A, but includes some furniture: an all-in-one desktop, a printer, a desk, two chairs with metal frames, and a few other smaller objects. The addition of furniture makes this case more similar to a typical apartment room in which this system may be deployed. So more multipath effects, reflections, shadowing, etc. are expected. Figure 13a depicts the room setup with likenesses of the larger furniture. Additionally, the dimensions of the room are 11.5 ft x 12 ft (3.5 m x 3.7 m).

Room C: Apartment Bedroom.

Room C is also similar in construction materials as Rooms A and B. However, instead of mainly metal objects, the furniture in Room C consists largely of absorbent materials including a queen-sized bed, pillows, stuffed animals, books, and clothes. As a result, more absorption and attenuation of signals is expected in this room than the others.



(a) Room B (b) Room C

Figure 13. Floor plans of Rooms B and C including larger furniture items

Figure 13b depicts the room setup with likenesses of the bed and bedside table. The dimensions of the room are 12.4 ft x 12.9 ft (3.8 x 3.9 m).

3.5 Limitations

The approach towards the proof of concept has been mostly developed now with assumptions, equipment, experimental methods and evaluation metrics, and testing scenarios. However, certain limitations still exist. For example, Insteon devices do not constantly transmit. Without a field of RF signals to exploit, the event of human presence will be difficult to detect. Also, RSS is not recorded by Insteon devices, so SDR receivers were chosen as an alternative. However, how these SDRs operate in the proof-of-concept system is still to be discovered. Therefore, additional preliminary work is needed before the proof of concept can be deployed.

IV. Preliminary Work for System Development

The previous chapter presented approaches to testing the RSS DfP detection and localization system, but additional preliminary work is needed to develop the actual proof of concept. For example, a way to get an Insteon network to transmit signals to be exploited must be developed. Therefore, this chapter pursues the development of a method for generating Insteon messages on a prompt and the development of data collection and processing algorithms and scripts for eventual detection and localization in the next chapter. The work in this chapter spans various subjects, including protocol sniffing, reverse engineering, security analysis, and code development.

4.1 On-Demand Generation of Insteon Message Fields

The proof-of-concept system exploits existing Insteon RF signals in the area of interest. Thus, this system requires the home automation network to transmit signals when it wants to use detection or localization. As with most home automation devices though, Insteon devices do not continuously transmit signals. These silent periods are not exploitable, but it would be prudent to investigate how the proof-of-concept system can generate these signals when it wants to perform either detection or localization.

4.1.1 Preliminary Investigations.

The goal of this section is to investigate how to generate signals to reduce these non-exploitable silent periods. There is ideally a method that completely eliminates the silent period by overriding the Insteon protocol that synchronizes messages on the zero-line crossing (see Section 2.5.1). However, the reduction of silent period should be sufficient for purposes of this proof of concept. The reduction of the silent period can be achieved

by prompting Insteon devices to send messages on demand. Therefore, two questions are investigated for this section: how to prompt Insteon devices to send messages on demand and which message to send.

Conditions.

The generation of signals should be on demand for when detection or localization is desired. In other words, only when detection or localization is about to be carried out does the signal generation occur. This saves power consumption costs and reduces the disruption of normal use of the home automation network.

The signals should contain authentic Insteon messages. In other words, actual Insteon devices should be able to collect and process these messages. If the message contains a command, then the device should also carry out the command. This prevents the unnecessary use of a device operating on the 915 MHz band without any other purpose. This also promotes the generation of additional signals due to the message relay protocol discussed in Section 2.5.1.

The generated signals should not alter any device physical state. For example, signals should not change the light status of a device from 'On' to 'Off,' change brightness levels, or put the device into the linking state. This condition should keep the system discrete and unobtrusive.

The signals should transmit from an Insteon device. In other words, Insteon devices themselves should send out these signals, not a separate-party rogue device operating on the 915 MHz frequency and capable of crafting their own Insteon RF messages. Again, this continues the idea that this system exploits an already-existing network of devices. Also, Insteon devices relay messages only to and from other Insteon devices, so this condition also increases the number of signals in the network.

Reverse Engineering the Hub App.

The first investigation attempts to find out how to exploit Insteon devices to send messages on demand. As evidenced in Section 2.5, information about general Insteon specifications and design are generously provided, but to generate Insteon signals for exploitation, the proof of concept needs to investigate non-intended uses and possibilities of the devices and their protocols. In fact, Section 2.5.3 noted that no Insteon documentation exists concerning security, especially for Insteon Hubs and the associated software applications. Internet connectivity is a popular avenue for exploiting IoT devices. Therefore, the Insteon Hub and its app are promising for these purposes.

The user interface of the mobile app is explored first. This method of using the official app requires only a device capable of running a version of the "INSTEON for Hub" app and capable of connecting to the Internet. Using the app showed that response time is often very slow and irregular between sending messages. Up to four seconds of wait time were observed since the app waits for a status update from the device and must refresh before another button could be pushed to send another message out. Rarely would an instant (< 1 s) response occur. Therefore, using the app as it is intended would not generate enough signals for clear detection and localization using RSS, so more investigation is necessary.

Instead of going through the app to generate messages, finding out how the app sends commands from the Internet to the Hub could eliminate the long, in-app waiting time. Documentation on the Insteon Application Programming Interface (API) exists [56]. While the API does say that the commands are sent by HTTP and gives information on what commands can be sent, the information is API-specific and requires items like API keys. According to the Insteon website, getting an API key is reserved for developers and "strategic partner." Pursuing this is beyond the proof of concept, but it is discussed in Section 6.3 for future development. Hence, the investigation continues.

Mobile apps are often written by developers and contain plenty of information about how the device functions. Reverse engineering a mobile app allows non-developers to look at the code and therefore learn more about the app and how it functions. To reverse engineer the "INSTEON for Hub" app, the same version of the app was downloaded from APKPure [57]. After extracting the compressed file, the Android Package (APK) was converted into a Java Archive (JAR) file. Then to display the Java source code of the JAR file, a Java decompiler was used. The result was an easy-to-read and -navigate source code with intuitive strings and class names in a search-capable graphical interface.

The main objective of reverse engineering the Hub app was to find out how the app sends messages to the Hub. Therefore, the keyword 'send' was first searched in the decompiler. The result was 189 items, but the `SmartLincManager` class was the most relevant. In the class, commands like `sendCommand` and `sendWebCommand` were found. Interestingly, a few of the commands had arguments of strings of the same format as the serial communication protocol discussed in Section 2.5.2.

The search query 'http' produced much clearer results in the same class. For example, there were strings that created an HTTP GET message with an Authorization field. There was also a command, `httpRequest`, that asked for a string of the Hub address, a double of the port number, and a string of the command. Cases for various commands were also seen. Based on these findings, the `SmartLincManager` class sends messages from the app to the Hub using the serial communication protocol.

Therefore, the approach to pushing Insteon devices to send messages appeared to be to use the Insteon Hub and prompting it to send messages by interfacing with it through its serial communication protocol. Such a task is simple and can be achieved using any device capable of establishing a HTTP connection. For simplicity, an Internet-connected laptop will be used for further investigation of this on-demand message generation.

Candidates for a Command to Send from an Insteon Device.

With an idea as to how to prompt Insteon devices to send messages more quickly than through the mobile app, the next part of the investigation attempts to determine what type of message would be ideal to send to reduce silent periods. Section 2.5.1 discussed the protocol for the three different types of Insteon messages. The Broadcast message is not a candidate, because it is often used for linking devices, which would disrupt normal use and thus violate one of the above conditions. Device and Group messages, on the other hand, have more uses.

According to the API, the only valid commands for Group messages are 'On,' 'Off,' 'Start Dim Up,' 'Start Dim Down,' and 'Stop Dim'. Direct messages have other possibilities for the different types of devices. For instance, a dimmable device like the Keypad or LED bulb can be validly sent 'Get Status,' 'Beep,' or 'Instant On' in addition to the Group commands. The issue with the Group message possibilities is that they all could violate the physical state alteration condition. 'Get Status' is a prime candidate since the command only asks the device to send back the light status. 'Stop Dim' is a secondary candidate since it only affects the physical state if the devices are currently dimming up or down, which is not common.

Other than looking at the possible commands, the protocol for either Group or Direct messages is a factor. With Direct messages, the protocol is short and simple; the originating device sends out the message, and the destination device receives it and sends back an acknowledgement. Group messages, however, have a longer protocol; the originating device sends out the Group messages, then follows up with individual messages to each of the group members asking for an ACK. Each of the devices then sends back an ACK. So while the Direct message protocol only sends two messages, the Group message protocol sends $2G+1$ messages, where G is the number of members in the group. This assumes no repetition, so these numbers are minimums for perfect transmission.

So while the Direct message has the prime 'Get Status' command, it only expects two messages sent out. And while the Group message can expect $2G+1$ messages for just one command, its best command is 'Stop Dim,' which has a small chance of disrupting the user's control. Ref. [53] also suggests that 'ID Request' is a better example than 'Get Status' because the response is two messages: an ACK and then a message with the Insteon ID.

Testing HTTP Serial Communication in a Browser.

Reverse engineering the Insteon app revealed that communication to the Insteon Hub from the app uses HTTP requests based on the serial communication protocol. Thus, further investigation into the protocol described in Section 2.5.2 should help design the HTTP requests for message generation. To investigate, URLs are crafted for different commands to send through a web browser. Wireshark is additionally used to sniff the HTTP packets being exchanged.

As previously mentioned, the two commands of interest are the 'Get Status' Direct command and the 'Stop Dim' Group command. Since these commands do not create any visual changes in the home automation network, the Hub buffer is also checked to verify that the messages are sent and acknowledged. The Hub's IP address and port number can be found at `https://connect.insteon.com/getinfo.asp`. This webpage checks Insteon's online database of devices and returns devices that have been found on the current network until the device changes account ownership or is removed from the database by manual request. Information on this page includes the link to the Insteon device, the user's network IP and gateway, and some Hub device specifics. Based on this information, the four preliminary URLs to test are listed in Table 7.

Table 7. URLs for testing the Hub's serial communication protocol

Description	URL
View Insteon buffer	http://10.1.0.26:25105/buffstatus.xml
Clear Insteon buffer	http://10.1.0.26:25105/1?XB=M=1
'Get Status' direct message	http://10.1.0.26:25105/3?0262347cfd0F1900=I=3
'Stop Dim' group message	http://10.1.0.26:25105/0?1805=I=0

To use the URL commands, the request must come from an address on the same network as the Hub. As expected, when the URL to view the buffer was accessed, authentication was prompted and required a username and password. The default credentials are on the back of the Insteon Hub and seem randomly generated without any database of possible combinations found anywhere on the Insteon site. These credentials can be changed by the user with the app, but the app does not force this change. The monitored HTTP packet showed that the authentication realm is basic protected and that the credentials are base64-encoded from `username:password` format. Successful access of the buffer showed residual data, so the buffer was erased, producing all 0's. Figure 14 shows a Wireshark example of the sniffed HTTP packet of a successful attempt at accessing the buffer.

The first URL command tested was the 'Get Status' Direct message. The attempt was successful; in the buffer was

```
0262 347CFD 0F1100 06 0250 347CFD 392182 20 00FF
```

which, according to documentation, is the appropriate output. The buffer is space-delimited here for better explanation. The first three segments echo the message sent. The following segments are the response. For the status request of Keypad 1 when turned completely on to maximum brightness, the response was that the message from `34.7C.FD` to `39.21.82` was a Direct message acknowledgement (`20`) with the response `00FF` indicating the status. If the device was off when the status was requested, then that last segment was `0000`.

Figure 14. Screenshot of Wireshark capture packet containing HTTP GET and all header fields

Next tested was the 'ID Request' Direct message. The attempt was successful; in the buffer were three different segments: the command echo, the first ACK, and the ID response.

```
0262347CFD0F1000 060250347CFD392182201000 0250347CFD0142448F0176
```

Then was the 'Stop Dim' Group message. It is unclear if the attempt was successful; the buffer was not as expected from the documentation:

```
0262 000005CF 180006
```

Other Direct commands, like Status Request, were attempted for Group messages, but the buffer did not update. These results suggest that only the Direct commands can be used for the proof of concept. Nevertheless, it would be interesting to see if the Hub emits a signal for these commands to see if there are other possibilities than what was written.

Viewing Signal Spectra of Different Commands.

Using the SDR, the spectrum analyzer in MATLAB® revealed a lot of information:

1. The Hub did send out a signal for 'Stop Dim' Group messages, and it was one long signal. Ref. [53] says that certain controllers do not send out a Group Cleanup after this command.
2. Different signal lengths (like in ref. [49]) were observed even when sending the same command.
3. Direct messages produced one message, but signal lengths were never as long as that of the 'Stop Dim' Group message.
4. 'Stop Dim' Group messages to larger groups did not increase the length of the signal.
5. Valid, visible commands like 'On' or 'Fast Off' to a group produced more than one signal. For a group of two devices, one long message followed by two short messages and then a long message was observed. For a group of three devices, it was long, short, short, short, long; and so forth for bigger groups. This corresponds with Group Cleanup protocol.
6. Four additional long signals were sent out when sending a visible command to a group with a missing device.
7. Only one signal was observed when sending a single-message Group command with a missing device.
8. One long signal was observed when sending Direct commands to a device not plugged in but still previously linked to the Hub.
9. Three medium-length signals were observed when sending a Direct command to a device plugged in but not previously linked to the Hub.
10. Five long signals were observed when sending any Direct command to a device neither plugged in nor previously linked to the Hub.
11. For any command with more than one signal, there are pauses between signals. This is most likely due to zero-line crossing synchronization.
12. There appears to be a small buffer for messages prompted while messages are currently sending; after refreshing the URL many times, signals still appeared even after the refresh stopped.

4.1.2 Message Generation Script and Test Use.

Appendix A contains the entire Python 2 script for generating Insteon messages to send RF signals when prompted. The above investigations directed the script to use HTTP GET requests to prompt the Hub to send out signals. The investigations also showed that a Direct message to any unlinked device (e.g., `00.00.00`) with any command (e.g., `0x1000`) is the best message to send that follows the conditions. With just one message, five long signals were observed in the spectrum analyzer. This reduces the number of message generations required, which would have consumed more power and accrued greater computational costs.

To further lower costs, the minimum number of bytes to send over HTTP is desired. After trial and error, only two of the header fields from Figure 14 were found to be necessary: Cache-Control and Authorization, in that order. Therefore, the entire HTTP GET request has the following format:

```
GET /3?02620000000F1000=I=3 HTTP/1.1\r\n
   Cache-Control:  max-age=0\r\n
   Authorization:  Basic <base64 credentials>\r\n
   \r\n
```

where the base64-encoded `username:password` could change. Credentials are user-inputted as an argument or found in a file called "prevCreds," which is not secure, but not an issue for this proof of concept.

This request is sent through a socket. The socket is connected to the Hub's IP address and TCP port and sends the request. If successful, the connection would close such that one message is sent over one connection.

To get the Hub IP and Port, three different methods were written into the script: file parsing, user-inputted argument, or parsing `http://connect.insteon.com/getinfo.asp`. The script checks a file, "prevHubConnect" for a valid address/port combination.

Table 8. Arguments for the message generation script

Argument		Description
-s	–src	Hub IP address and TCP port number as `ip:port` Default: checks file then web page
-a	–auth	Authorization credentials for Hub as `username:password` Default: checks file
-c	–count	Number of times to send message Default: infinite loop
-t	–time	Delay in s between sending messages Default: 0.5 s

This can be bypassed if the script is called with an argument specifying the address and port. If a valid destination is not found, then the script scrapes the above URL.

The requests are sent in a loop, whose length can also be determined by an argument parameter. By default, this is an infinite loop that is broken if the process ends. Since this script is designed to be run as soon as data collection begins, then the process ends when data collection ends. Alternatively, the argument parameter can designate the number of messages to send before quitting.

The requests also are time-delayed. The default is 0.5 s, but this can be changed by an argument as well.

By default, the script finds the Hub's IP address and port number, retrieves the authorization credentials, and infinitely loops to send the same message over HTTP every 0.5 s. Table 8 lists all the arguments that can be passed into the script and their effects. Figure 15 presents examples of calling the function and the console outputs. As verified by the spectrum analyzer, the script generates the same signals as expected from the investigations.

The script is designed to run from the data collection scripts. Ideally, this process would not be necessary in a real system. Instead, the devices would have their own pro-

```
python InsteonMsgGeneration.py
Hub 392182 sent 1 message(s)

Hub 392182 sent 2 message(s)

Hub 392182 sent 3 message(s)

Hub 392182 sent 4 message(s)

Hub 392182 sent 5 message(s)

Hub 392182 sent 6 message(s)
```

(a) Default parameters, Hub found on website

```
python InsteonMsgGeneration.py -c 5
Hub Your Hub sent 1 message(s)

Hub Your Hub sent 2 message(s)

Hub Your Hub sent 3 message(s)

Hub Your Hub sent 4 message(s)

Hub Your Hub sent 5 message(s)
```

(b) Changed count parameter, Hub previously manually set

```
python InsteonMsgGeneration.py -c 5 -a bad:creds
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Basic realm="Protected"
Connection: close

401 Unauthorized: Password required
```

(c) Set authentication parameter to bad credentials

Figure 15. Examples of console outputs for message generation script

tocol for sending RSSI for DfP detection or localization. However, for the proof of concept, it is sufficient to run the script from a second laptop with an Internet connection and a Python compiler.

4.2 Data Collection and Processing Algorithms

Ideally, Insteon devices would send RSSI to each other. If this were true, the the proof-of-concept system would only have to sniff these RSSI. However, it was discussed in Section 2.5.1 that Insteon messages do not have a field for carrying RSSI. Without this capability, the system needs an alternative to perform RSS-based DfP detection and localization. This alternative is NooElec receivers (see Section 3.2). This section discusses the scripts and algorithms for detection and localization that make use of the SDR receiver.

All data collection and processing scripts are written in MATLAB® because there was a support package for the NooElec receivers. Separate algorithms are developed for detection and localization because of their different approaches and setups, as discussed in Section 3.3. However, several aspects should remain the same. One such aspect is that RSS calculation during data collection is the same in both methods.

Table 9. SDR configuration parameters for MATLAB® collection

Parameter	Value
Tuner frequency	915 MHz
Tuner gain	45 dB
Sample rate	2.4 MHz
Samples per Frame	2560

The SDR collects signal data. Table 9 shows the collection parameters set in MATLAB® for the SDR. The length of a collection is set by a predetermined simulation time in seconds. Data from every collected frame is converted into RSS in dB using (11). The equivalent MATLAB® command to convert to RSS from the amplitude collected from the SDR frame data, z, is:

```
20 * log10(abs(freqz(z,1,ω)))
```

where `freqz`$(z,1,\omega)$ is the frequency response of the digital filter and $\omega = 2\pi\frac{x}{N}$, where N is the frame length and $x \in [-\frac{N}{2}, \frac{N}{2}-1]$.

Using this method, a frame that collected a 915 MHz signal would have two peaks. Therefore, for every sample, the largest two values captured in a frame are averaged together and recorded as the single RSS value, R, along with the time that this frame was captured. Additionally, the first step collected was observed to always be an outlier, most likely because of device initialization, and so is ignored for every collection.

4.2.1 Detection with a Single Node.

Section 3.3.1 presented the method for detection in a simple tripwire scenario. There are two main phases: setup and calibration, and online data collection and processing. Similar to Figure 8, Figure 16 reflects these phases as well as the algorithm for determining a detection event during an experiment.

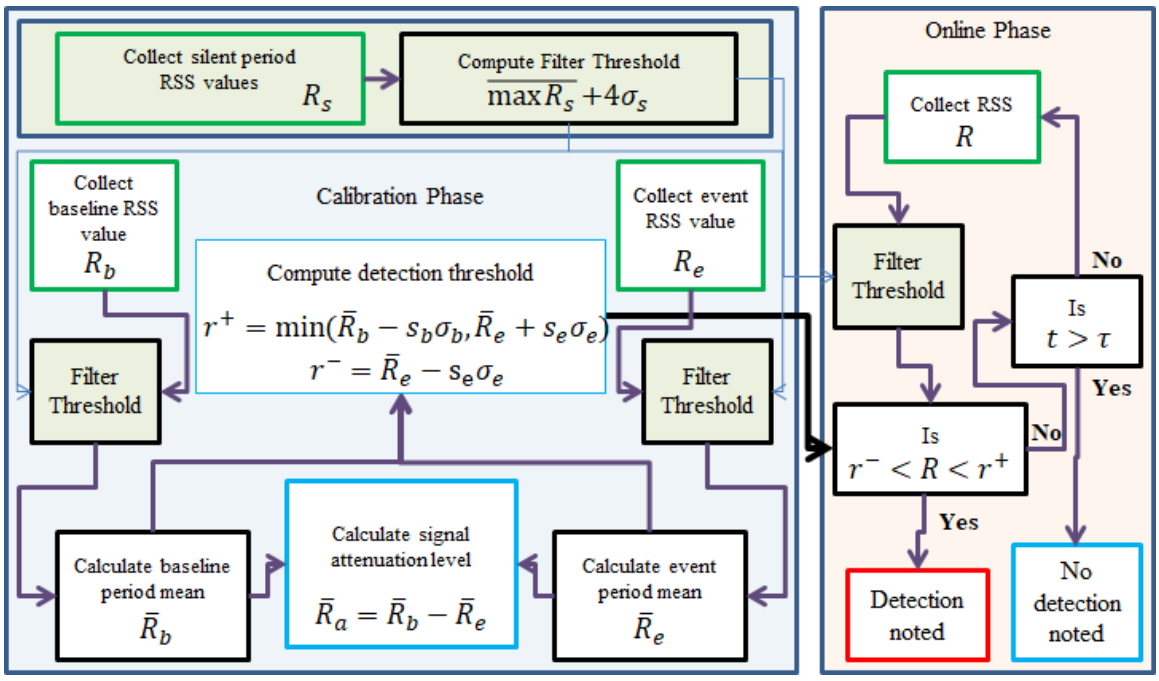


Figure 16. Flow chart depicting the algorithm for a detection event

Calibration Script for Detection.

The left-hand side of Figure 16 depicts the calibration phase for detection. As mentioned in Section 4.1, the message generation script is only able to reduce the silent periods of the Insteon protocol, not eliminate it. Therefore, the largest difference in the calibration phase of Figure 16 compared to that of Figure 8 is the inclusion of a silent period calibration and filter threshold.

Let the s subscript denote a silent period when neither the target nor the 915 MHz signal is in the area of interest. From a preliminary analysis of the mean silent period RSS, $\overline{R}_s$, was much less than either $\overline{R}_b$ or $\overline{R}_e$ (about 14 dB vs. about 60 dB). However, single values of R_b (without the target) were observed as low as 20 dB. Such results would cause a very large standard deviation, and since the method uses a deviation-dependent threshold for detection, this could form a large threshold with a very high probability of false positives. Therefore, the silent period needed to be filtered out for collection in both calibration and online phases.

However, using the mean, $\overline{R}_s$, by itself would not eliminate these outlier values of R_b since $\overline{R}_s$ was less than these observed minimum values of R_b. Another preliminary observation of the silent period was that the collections sometimes observed a few values of R_s that were much higher than the mean, ranging from 20-50 dB. Since this max R_s was still less than $\overline{R}_b$, a filter threshold using the average of max values instead of the smaller mean could be beneficial for reducing outlier data not reflecting typical behavior. At least for detection, this could be very important.

Definition 6 *Let* $\overline{\max R_s}$ *be the average maximum* R_s *value observed in several collections of the silent period and* σ_s *be the standard deviation of all* R_s *in these collections. Then the silent period filter threshold,* f_s *is defined to be four times* σ_s *above* $\overline{\max R_s}$*:*

$$f_s = \overline{\max R_s} + 4\sigma_s \tag{30}$$

The detection experiment in Section 5.1 will determine if this filter threshold is sufficient or excessive. Note that localization may not use the same filter threshold since fluctuations in RSS should not be as big of a concern in fingerprinting as it is for baseline threshold detection.

In the calibration phase of the script for detection, computing this filter threshold is the first step. In a real scenario, calibration of the silent period should not need to occur more than once. Suppressing the network signals for a short period to calibrate this silent period should not be too much of an inconvenience anyway. Regardless, the experiments in this research simply calibrate the silent period once, before calibrating the baseline and event periods.

Baseline and event periods are calibrated for every distance d in the detection experiment. In a real scenario, these periods would need to be re-calibrated every time there is a large change in the area, such as new furniture. However, that is not a concern for

providing a proof of concept.

The script for calibration after the silent period filter threshold is computed then begins calibration of the baseline period. The user is prompted to begin the calibration and when the process is completed, $\overline{R}_b$, σ_b, and n_b are calculated after running all R_b through the filter threshold. Immediately following this calibration is the event period, after which $\overline{R}_e$, σ_e, and n_e are calculated similarly. In both cases, the message generation script should already be running, and for the event period, a target should be present at the d being calibrated.

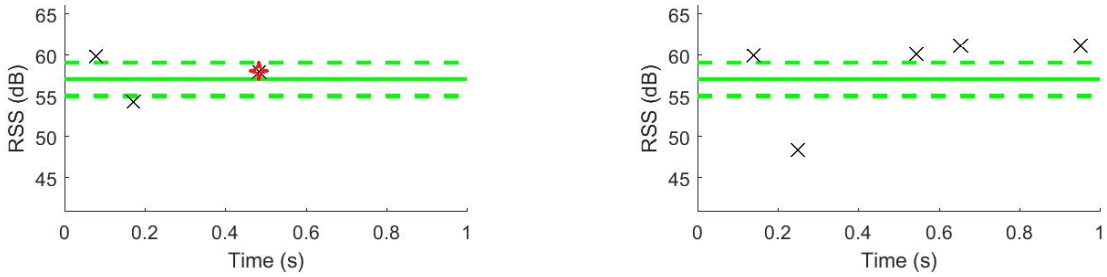
After both baseline and event periods are calibrated, the mean signal attenuation level, $\overline{R}_a$, is calculated using (12) and the detection threshold is computed using Definition 2. This threshold is then stored for detection.

Collection and Processing Script for Detection.

The right-hand side of Figure 16 depicts the algorithm for the online phase of detection. Detection is written to allow for live collection and determination of a detection event. No RSS values are recorded during detection; instead, every step above the filter threshold is compared against the detection threshold as per Definition 3.

Due to fluctuations in signal strength, it is not expected that every R for which there is a target will produce a detection result. Instead, it is more plausible to evaluate the detection event for a period of time. Let τ in detection be the time period in which a detection event must occur. If the baseline threshold is breached before this time limit, then a detection event is reported (Figure 17a). If the live time, t, exceeds τ, then no detection is noted (Figure 17b).

For the experiment, live collection occurs in a loop for some number of trials. In the line-of-sight case, the user is also prompted enter a distance d in order to use the appropriate threshold. The distance tested is assumed to have been previously calibrated.



(a) Detection noted (RSS in threshold)

(b) No detection noted (timed out)

Figure 17. Examples of live collection plots for detection with baseline threshold range as dashed green lines

4.2.2 Localization with Multiple Nodes.

The cell-based fingerprinting method is covered in Section 3.3.2. Similar to detection, there are two main phases, but these are referred to as the "offline" and "online" phases. Figure 9 in Section 3.3.2 depicted how the localization method works.

Offline Calibration Script for Localization.

In addition to the program-based calibrations, users should also note the cell divisions in the area of interest. Although there is potential for having a room builder that does this within the system, for this proof of concept, the area of interest is manually drawn up, with one example in Section 3.4.2. The cells should be labeled consecutively from 1 to C.

The calibration script first asks for this number of cells so that it can allocate its arrays. The number of nodes, N, is sensed by the program (using `sdrinfo`) for the same purpose. Once both arrays are allocated, the script then asks the user to calibrate the cells, one at a time. During calibration, a user first calibrates the empty case, when there is no human presence in any cell. After which, a target should be inside the cell while the signal is being emitted for cells 1 to C. For each case, the program records all R_e above the filter threshold, computes $\overline{R}_e(c,n)$ from each node and stores it into the matrix $\overline{\overline{R}}_e$.

(It also computes and stores $\sigma_e(c,n)$ and $n_e(c,n)$.) When all $C+1$ cells are calibrated, live collection may begin.

Online Collection for Localization.

The online phase script is similar to the live collection phase script for detection in that they are both time-limited programs. However, the fingerprinting method is much more complex than the baseline comparison method. The system must scan a short window, τ, of the samples and use the probability-based fingerprinting method in order to select the probable location of a target. For developmental purposes, the proof of concept separates live collection and processing from the actual analysis.

During live collection for the proof-of-concept experiments, the script loops for the number of trials, and the user is prompted to begin each trial. The signals above the filter threshold from all nodes $n \in \{1,2,...,N\}$ are collected and processed for mean RSS as a function of time. For every window beginning at $t_1 = t_0 + m\tau$, the results are stored as a vector of $\overline{R}(t_1,n)$ values ($\sigma(t_1,n)$ and $n_o(t_1,n)$ are also included). The vectors from all trials are then saved for later analysis.

Online Analysis Script for Localization.

In the analysis script, the program loads the saved data from each trial and performs the probabilistic selection algorithm for the windows of samples. If a window's recorded values from the N nodes do not probably match the $\overline{R}_e(c_0)$ vector, then a cell $c \in \{1,2,...,C\}$ is selected as the most probable location of the target, as per Definition 5. The window time and cell location can both be printed to the console for each localization result returned, but these are just stored in arrays for analysis.

A possible difference between this script and the one for detection is that the silent period filter threshold may be set differently. A big difference is that this script involves

multiple nodes simultaneously collecting. As a result, the algorithm is slightly more complicated and the computational complexity is greater. This led to the separation of the collection and location analysis phases for this proof of concept. Therefore, several limitations exist for localization as-is, but they are acceptable for a proof of concept.

4.3 Summary

The DfP system requires RF signals to exploit, but Insteon devices produce silent periods when they do not need to transmit. Therefore, the message generation script was developed to send these messages over HTTP to the Insteon Hub. Investigations showed that the Insteon app communicated to the Hub using HTTP requests. As a result, URL commands were attempted in a browser to verify that this works and to sniff the HTTP packet of a successful connection. Several different commands were attempted for both Direct and Group Broadcast messages. Although the Direct message for 'ID Request' was a prime candidate for generating signals discretely for DfP purposes, spectrum analysis revealed that a more general Direct message to an unlinked or improbably device ID like `00.00.00` always results in five long signals, which reduces the frequency that a connection must be established for a message to be sent.

Since Insteon devices do not transmit or receive RSSI in their messages, the proof of concept needed an alternative for collecting RSS values. The result was affordable SDR receivers that could be integrated into currently-existing devices. Scripts were written to collect and analyze RSS using these receivers for either detection or localization.

With scripts for message generation, simple setup detection, and cell-based localization written, the proof-of-concept system has been developed and can be deployed for detection and localization.

V. Experiment Results and Analyses

This chapter presents results from the deployment of the RSS DfP system in order to test the proof of concept's detection and localization capabilities and to evaluate the performance of the system for future use and development. Section 5.1 presents work to verify that the target attenuation of the 915 MHz signal is significant for detection and localization using the system. Results from detection at the midpoint and then along the LOS of a simple tripwire configuration are also presented. Sections 5.2 and 5.3 contain experiments seeking to demonstrate that fingerprinting-based localization is possible. Section 5.4 presents and compares results from localization performance in different rooms to evaluate the robustness of the system.

5.1 Detection with Simple Baseline Threshold Comparison

The primary question for the proof of concept is whether Insteon devices emitting 915 MHz signals can be used for RSS-based DfP detection. The experiments presented in this section attempted to answer this question by using the proof-of-concept system to first establish that the foam blocks, as an approximation of the human body, does attenuate the signal significantly enough for detection. When this was shown, the system could then be used to perform detection of a target at the midpoint of the tripwire. The same principles could then be applied for detection of a target located at other points along the LOS of the two devices.

The experimental setup was discussed in Section 3.4.1 and visually depicted in Figure 10. Sections 2.2.2 and 4.2.1 explained the methods and algorithms for detection used in the system.

Table 10. Results for silent period calibration for detection in Room A

Trial	n_s	$\overline{R}_s$ (dB)	Max R_s (dB)	σ_s (dB)
1	854	13.9	22.4	1.5
2	859	13.9	32.5	2.3
3	858	13.9	18.4	1.3
4	857	13.8	40.5	2.6
5	860	13.8	42.2	3.3
6	856	13.7	46.1	2.6
7	862	13.7	28.3	1.6
8	861	13.7	22.5	1.8
9	857	13.9	24.6	1.4
10	873	14.3	49.1	4.6
11	872	14.0	27.7	1.4
12	869	14.2	41.6	3.2
13	876	14.2	44.6	3.0
14	864	14.3	48.7	4.9
15	869	14.6	29.4	1.8
Avg		14.0	34.6	1.6

5.1.1 Signal Attenuation and Calibration at the Midpoint.

The first step to demonstrate that the system functions was to successfully capture signal attenuation levels at the midpoint in Room A. For analysis, the calibration script was separated into the three different periods: silent, baseline, and event. First, the silent period was calibrated to view what ambient noise was picked up by the receiver nodes. This silent period could then be removed from the other collections. Fifteen 60-second collections were recorded without a signal and without a target present. The results from all fifteen collections are shown in Table 10.

Approximately 860 steps for each 60-second trial were recorded using the receiver nodes. The variance was high when a few stray signals much greater than the mean were detected (e.g. trials 10 and 14). As per Definition 6 in Section 4.2.1, this data was used to construct a silent period filter threshold. The weighted average of the maximum R_s among all 12,947 samples was 34.6 ± 1.6 dB. Therefore, the silent period filter threshold

was such that any values under $f_s = \overline{R}_s + 4\sigma_s = 41.0$ dB were ignored in the baseline, event, and detection period collections to reduce outliers from silent periods.

With f_s calibrated, the baseline and event periods were then collected for threshold calibration. These collections were paired; for each trial, the event period was collected right after the baseline period. Fifteen trials were conducted for each period, each for 60 seconds. Despite having the same time length of capture as the silent period trials, these results had only about 370 steps each because of the filter threshold. These results, along with paired signal attenuation levels calculated using (12), are shown in Table 11. Figure 18 shows the boxplots summarizing the collected signals from each period for each trial. The mean signal attenuation level from a single target at the midpoint of Room A was $\overline{R}_a = 4.0 \pm 0.8$ dB.

A one-tailed, paired t-test was used to confirm that $\overline{R}_b$ and $\overline{R}_e$ were significantly different to allow experiments for detection at the midpoint (see Metric 1). The test results are shown in Table 12. The p-value was less than 0.0001, so the mean difference, $\overline{R}_a$, is statistically significant. This means that detection at the midpoint could be performed.

To avoid needing to pair baseline and event collections in future detection calibrations, the difference of $\overline{R}_b$ and $\overline{R}_e$ was evaluated and compared against the paired difference. Over 5537 samples, $\overline{R}_b = 61.1 \pm 1.3$ dB, and over 5512 samples, $\overline{R}_e = 57.1 \pm 1.0$ dB. This resulted in a difference of 4.0 ± 1.7 dB. The paired difference was not significantly different from this difference of $\overline{R}_b$ and $\overline{R}_e$. Therefore, pairing baseline and event periods would not be required for future detection calibrations.

The calibration values used for computing the calibration filter threshold and the baseline detection thresholds are summarized in Table 13. For detection at the midpoint in Room A, the threshold was initially set to $57.1 \pm 2(1.0)$ dB. Statistically, the range of $\pm 2\sigma_e$ should produce a small P_F while still detecting presence, even in a short time frame for detection.

Table 11. Results from paired calibration of baseline and event periods at the midpoint in Room A

Trial	Baseline Period, b			Event Period, e			$\overline{R}_a$
	n_b	$\overline{R}_b$ (dB)	σ_b (dB)	n_e	$\overline{R}_e$ (dB)	σ_e (dB)	
1	361	61.1	1.4	367	56.8	0.8	4.3
2	371	61.1	0.8	362	56.8	0.9	4.3
3	363	61.1	1.3	368	55.2	0.8	5.9
4	371	61.1	1.3	370	56.9	1.1	4.2
5	375	61.0	2.0	364	56.9	1.1	4.0
6	367	61.2	1.1	362	57.0	0.9	4.2
7	363	61.2	1.1	359	57.0	0.9	4.2
8	370	61.2	1.1	369	57.0	1.1	4.3
9	372	61.1	1.9	360	57.0	1.2	4.1
10	374	61.2	1.4	363	57.1	1.0	4.1
11	375	61.2	1.4	380	57.2	1.2	4.0
12	364	61.3	1.3	376	57.4	1.1	3.9
13	370	61.0	1.3	379	57.7	1.0	3.3
14	364	60.2	1.3	365	57.8	1.1	2.4
15	377	61.2	1.3	368	58.1	1.2	3.1
Avg		61.1	1.3		57.1	1.0	4.0

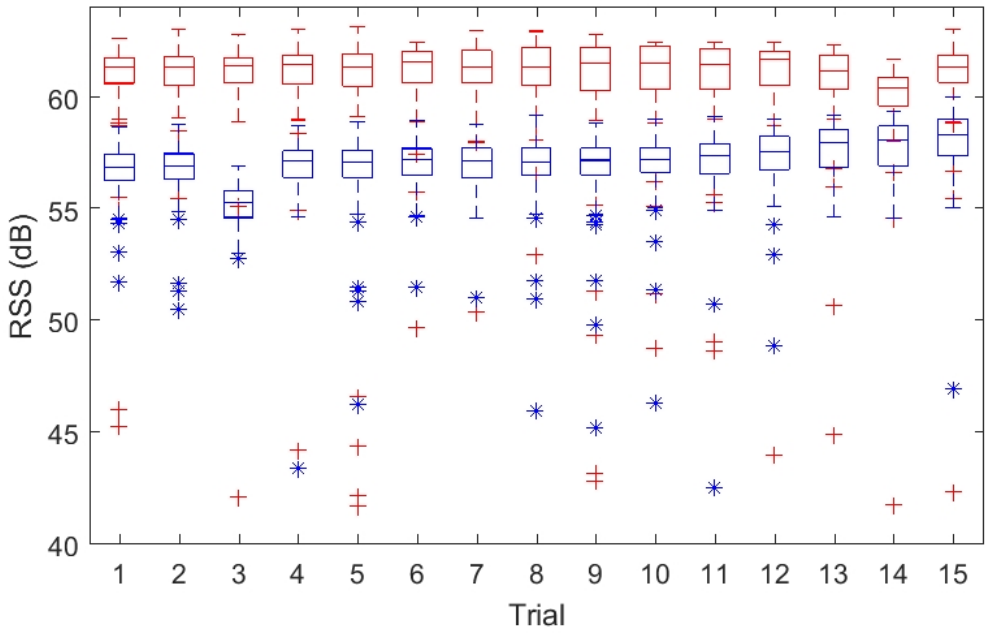


Figure 18. Boxplot of paired calibration trials of baseline (red, +) and event (blue, *) periods

Table 12. Results from the one-sided, paired t-test of $\overline{R}_b$ and $\overline{R}_e$ for determining signal attenuation, $\overline{R}_a$, at the midpoint in Room A

hypothesis test result	1 (reject H_0)
p-value	< 0.0001
95% confidence interval	$[3.7 \text{ dB}, \infty)$
test statistic	20.5743
degrees of freedom	14
mean difference	4.0 dB
standard deviation	0.8 dB

Table 13. Calibration values for detection at the midpoint in Room A

Variable	RSS (dB)
Avg. Max Silent RSS ($\overline{\max R_s}$)	34.6
Std. Dev. of Silent Period (σ_s)	1.6
Avg. Baseline RSS ($\overline{R}_b$)	61.1
Std. Dev. of Baseline Period (σ_b)	1.3
Avg. Event RSS ($\overline{R}_e(5.0)$)	57.1
Std. Dev. of Event Period ($\sigma_e(5.0)$)	1.0

5.1.2 Detection at the Midpoint.

From calibration in the previous section, the detection threshold at the midpoint was set at $57.1 \pm 2(1.0)$ dB. A total of 1,000 trials were performed using this threshold. The target was not present in the room for 500 of these trials, but was positioned at the midpoint between the transmitter and receiver devices for the other 500 trials. Each trial collected signals for up to $\tau = 1$ s, which should be an appropriate time frame to allocate for detection. When detection was noted, the trial immediately terminated. Between each trial, the program paused for a randomly-generated time of up to five seconds.

The results from all 1,000 trials are presented in Figure 19 and evaluated in Table 14. True positives were detected throughout the threshold range. While outlier values were collected, they were not recorded as a detection event. For one-second trials, the probability of detection was favorable at $P_D = 100\%$. All true positive detection events happened within 0.5 s, which suggests that a one-second interval is sufficient for detection at the midpoint using the calibration values. The probability of false positives was $P_F = 5.6\%$. Most of these false positives occurred near the upper threshold, which suggests that the threshold range could be lowered in order to achieve a desired, lower P_F.

In fact, doing so achieved that very result. Two additional threshold ranges, $\pm 1.5\sigma_e$ and $\pm\sigma_e$, were similarly tested for 1,000 trials (500 with no target present and 500 with the target at the midpoint) each. Figure 20 presents both histograms of the collected RSS distribution from these experiments. The probability of detection and the probability of false positives from all three thresholds are displayed in Table 15. Also, the accompanying ROC graph is in Figure 21.

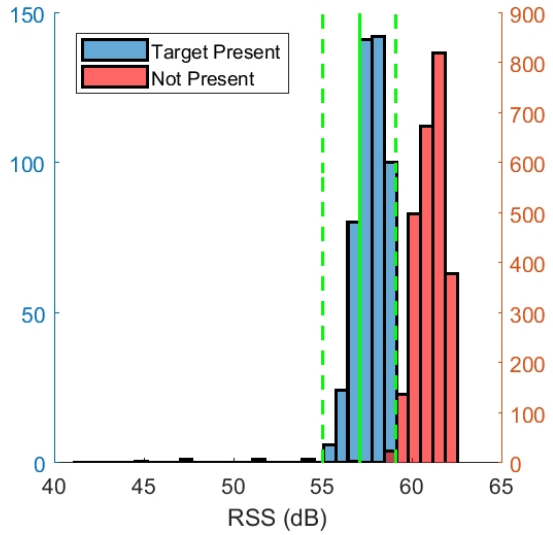
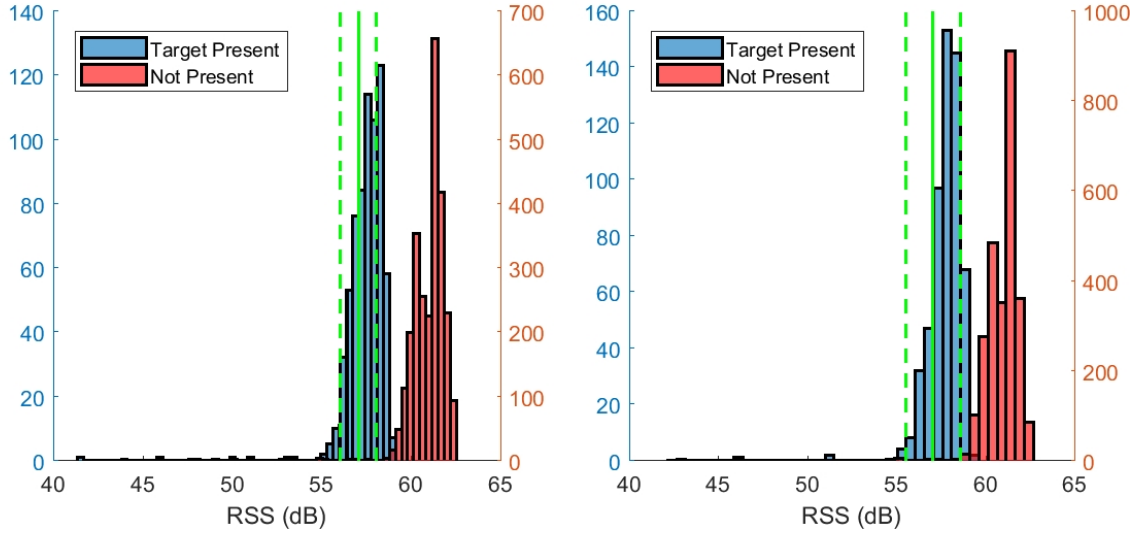


Figure 19. Histogram from midpoint detection trials showing the distribution of RSS collected before trial terminated due to detection event or time out



(a) $\pm\sigma$ (b) $\pm 1.5\sigma$

Figure 20. Histograms comparing RSS distribution from midpoint detection trials with two other threshold ranges

Table 14. Contingency table from detection at the midpoint

		Target Presence		
		Yes	No	Total
Detection	Yes	500 100.0%	28 5.6%	528
	No	0 0.0%	472 94.4%	472

Table 15. Comparison of P_D&P_F performance of three baseline threshold ranges for midpoint detection

Threshold	P_D	P_F
$\pm 2\sigma_e$	100.0%	5.6%
$\pm 1.5\sigma_e$	99.8%	1.6%
$\pm \sigma_e$	98.6%	1.0%

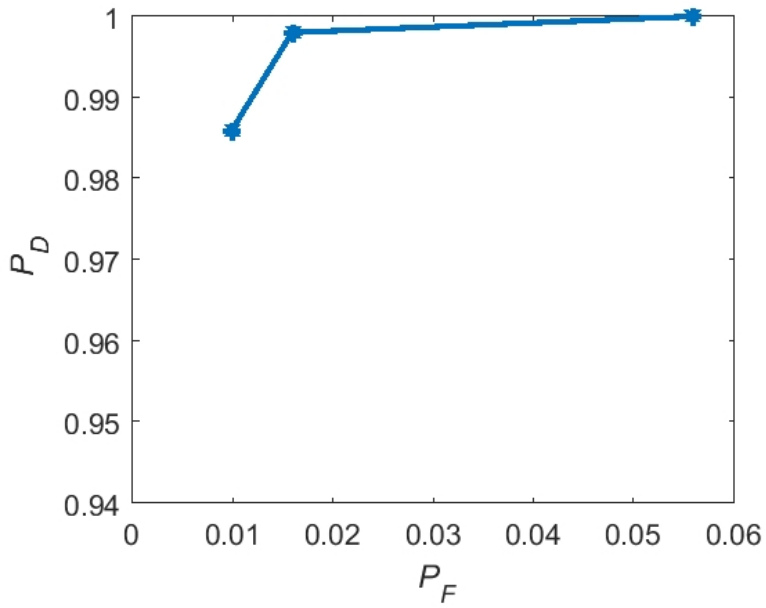


Figure 21. ROC graph for midpoint detection performance from varying the threshold range (points left to right: $\pm 1\sigma, \pm 1.5\sigma, \pm 2\sigma$)

Although $\pm 2\sigma_e$ had 100% probability of detection in one-second samples, the $\pm 1.5\sigma_e$ and $\pm\sigma_e$ thresholds performed similarly at P_D = 99.8% and 98.6%, respectively. These small differences are insignificant. However, by decreasing the size of the threshold, the changes to the probability of false positives were more significant (P_F = 1.6% and 1.0%, respectively). Therefore, the $\pm 1.5\sigma_e$ threshold was preferred over the $\pm 2\sigma_e$ threshold for the smaller P_F and over $\pm\sigma_e$ for the higher P_D. This threshold parameter (s_e = 1.5) was used in the rest of the detection experiment.

5.1.3 Line-of-Sight Calibration Thresholds.

To increase the functionality of the detection system, detection along the LOS was considered. Seven locations between the two devices were chosen: $d \in D_H$ = {1.0, 2.5, 4.0, 5.0, 6.0, 7.5, 9.0} ft (see Section 3.4 for metric conversions). The silent period calibration was reused from above since the environmental setup remained the same. The event calibration at d = 5 ft was also recycled. This left the event period to be calibrated for the remaining six locations along the LOS. As previously mentioned, separation of baseline and event calibrations was possible because paired calibration did not produce significantly different results from unpaired calibration. As a result, the baseline period calibrations were also recycled and used in calculating signal attenuation at different d.

The event at each d was calibrated with fifteen 60-second collections. Table 36 in Appendix B lists all of the summary statistics from each trial for all d. Table 16 below presents the resulting $\overline{R}_e(d)$ and the calculated $\overline{R}_a(d)$ for each d. Interestingly, $\overline{R}_a(d)$ did not increase as the target approached either device. Instead, the opposite was observed. At both ends, the mean signal attenuation was less than 2.4 dB compared to approximately 4.0 dB near the center. Also, $\overline{R}_a(2.5)$ was much greater than $\overline{R}_a(7.5)$ despite both being 2.5 ft away from a device.

One possible explanation is that when the target was close to either device, more sig-

Table 16. Calibration values for LOS detection at distance d from the Hub in Room A

d (ft)	$\overline{R}_e(d)$ (dB)	$\sigma_e(d)$ (dB)	$\overline{R}_a(d)$ (dB)	$\sigma_a(d)$ (dB)	t	p-value	H_0 result
1.0	58.7	1.1	2.3	0.3	104.43	< 0.0001	reject
2.5	56.6	1.0	4.4	0.6	196.99	< 0.0001	reject
4.0	57.0	1.1	4.0	0.8	173.24	< 0.0001	reject
5.0	57.1	1.3	4.0	0.8	175.24	< 0.0001	reject
6.0	57.3	1.1	3.7	1.1	160.85	< 0.0001	reject
7.5	58.6	1.1	2.4	1.2	107.27	< 0.0001	reject
9.0	58.9	1.2	2.2	0.1	90.11	< 0.0001	reject

nals were fully absorbed, so only slightly attenuated baseline signals reflecting around the room were actually detected. Since the target at 7.5 ft was farther from the transmitting device, the target did not block a lot of the transmitted signals, which would allow more signals to propagate and reflect around the room, around the target, before reaching the receiver. Another possible explanation is that the silent period filter threshold was too high for calibration closer to the devices, resulting in highly-attenuated signals that were not recorded.

Regardless, the baseline threshold detection method should still function at d if $\overline{R}_a(d)$ is statistically significant from $\overline{R}_b$. One-tailed, unpaired t-tests comparing $\overline{R}_e(d)$ to $\overline{R}_b$ were used to test this as per Metric 2. The results are presented in Table 16. In all cases, there was strong evidence to reject H_0, so detection could occur at all $d \in D_H$.

However, based on the thresholds in Figure 22, the performance of detection at some distances may not be favorable. In particular, d = 1.0, 7.5, and 9.0 ft had thresholds that overlapped the error lines for the baseline signal (shown as d = 0 in Figure 22). To help reduce the number of false positives that would accumulate for detection at these distances, the minimum argument described in Definition 2 for the threshold was implemented such that $r^+(d) = R_b - \sigma_b$, while $r^-(d)$ remained unchanged. The $r^+(d)$ change is reflected in the figure as the magenta error bars for d = 1.0, 2.5, and 9.0 ft. Results from LOS detection should answer how adequate this change was.

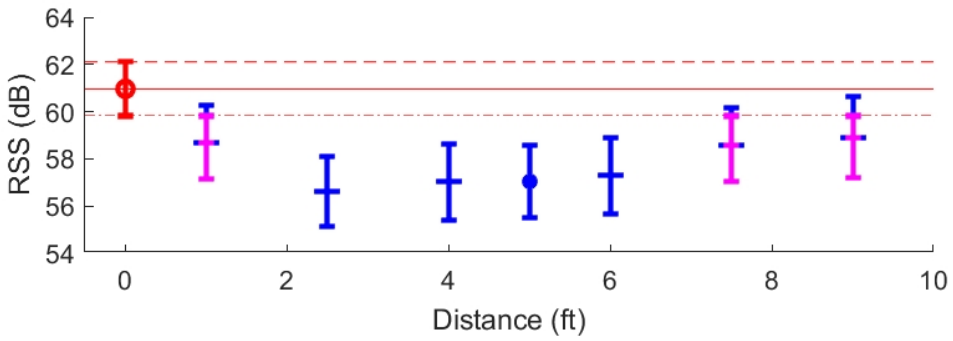


Figure 22. Error plot depicting $R_e(d) \pm 1.5\sigma_e(d)$ detection thresholds for LOS distances and alternate upper thresholds $R_b - \sigma_b$ for distances 1, 7.5, and 9 (magenta) compared to the baseline signal threshold $R_b \pm \sigma_b$ (Distance = 0, red)

5.1.4 Detection along the Line of Sight.

With the thresholds calibrated, experiments for detection along the LOS were conducted. Similar to the experiment at the midpoint, at each distance, 1,000 trials were performed, of which 500 trials contained target presence and 500 did not. Each trial was again one second long in duration. Figure 39 in Appendix B contains histograms showing the trial results for each of the d, not including the midpoint.

As summarized in Table 17, detection was possible at all points, but to varying degrees of success. The probabilities of detection were fairly successful ($P_D > 99\%$) at all distances except for right in front of the originating device (at d = 1.0 ft, P_D = 53.2%). This poor result can be attributed to the small threshold and the small attenuation value.

Closer to the two devices, the number of false positives was much higher. While target distances of 2.5, 4.0, 5.0, and 6.0 ft all resulted in $P_F < 6.5\%$, distances of 1.0, 7.5, and 9.0 ft respectively resulted in P_F = 53.2%, 51.0%, and 49.6% over 500 trials. These were with the alternate upper thresholds, $r^+(d) = \overline{R}_b - \sigma_b$. P_F should decrease with a smaller $r^+(d)$. However, further lowering $r^+(d)$ could also result in a lower P_D.

Table 17. Thresholds and performance (P_D, P_F) evaluation of LOS detection

d (ft)	Thresholds (dB) Lower	Upper	P_D	P_F
1.0	$\overline{R}_e(1.0) - 1.5\sigma_e$	$\overline{R}_b - \sigma_b$	81.8%	53.2%
2.5	$\overline{R}_e(2.5) \pm 1.5\sigma_e(2.5)$		99.8%	2.8%
4.0	$\overline{R}_e(5.0) \pm 1.5\sigma_e(4.0)$		100.0%	1.8%
5.0	$\overline{R}_e(5.0) \pm 1.5\sigma_e(5.0)$		99.8%	1.6%
6.0	$\overline{R}_e(6.0) \pm 1.5\sigma_e(6.0)$		99.8%	6.4%
7.5	$\overline{R}_e(7.5) - 1.5\sigma_e(7.5)$	$\overline{R}_b - \sigma_b$	99.8%	51.0%
9.0	$\overline{R}_e(9.0) - 1.5\sigma_e(9.0)$	$\overline{R}_b - \sigma_b$	99.6%	49.6%

Summary.

Motion sensors only work when the target moves, but RSS-based DfP detection can be used to detect human presence even when the target is stationary. This was demonstrated by detection experiments along the LOS of a transmitting home automation device and a receiving device. The detection experiments first calibrated thresholds for detection at different distances and then used these thresholds to test detection in a one-second time frame at each distance.

Mean signal attenuation levels were calculated for each distance using either paired or unpaired calibrations. Interestingly, the $\overline{R}_a$ one foot away from either device were much less than the level at the midpoint. Nevertheless, all signal attenuation levels were statistically significant from zero, with values at different distances ranging from 2.2 ± 0.1 dB to 4.4 ± 0.6 dB. Therefore, detection could be performed at all seven of the tested distances.

The experiment conducted at the midpoint showed that detection was possible, producing accurate detection and very few false positives. P_D and P_F were shown to vary depending on the threshold chosen, but $\pm 1.5\sigma_e$ produced adequate P_D and P_F in the 1,000 trials of one-second periods.

As expected, P_D from a baseline threshold method was high all across the LOS. In fact, all but d = 1.0 ft performed at P_D = 99.6% or higher in properly detecting a target within one second at that distance. The probability of detection at d = 1.0 ft was P_D = 81.8%, which was not entirely unfavorable, but the probability of false positives was high at P_F = 53.2%. Distances for which $\overline{R}_e(d) + 1.5\sigma_e$ did not overlap with the error in the baseline signal performed much better in detection and with much lower P_F.

The experiments in this section demonstrated that the proof-of-concept system can be used for detection along the LOS of two devices communicating using home automation signals. Limitations to this method were also shown, especially near either device. While simple-setup detection does perform, its setup, deployment and function are limited in real-life scenarios.

5.2 Localization with Cell-Based Fingerprinting

Localization with a multi-node setup has greater potential for deployment and use in real-life scenarios. It expands upon detection in that it can be used to detect targets and determine the target's location. By deploying multiple nodes, localization can determine target presence in a two-dimensional area of interest rather than just a one-dimensional LOS. The goal of the experiment in this section was to establish the functionality of DfP localization using the proof-of-concept system.

The method of localization investigated in this section was cell-based localization. This method was discussed in Section 3.3.2. To evaluate its performance, the experiment was conducted in Room A, described in Section 3.4.2. This setup differed from the detection experiment especially in that the area of interest was a 10 ft x 10 ft area with three receiving devices, the door was opened, and the DCPS laptop was inside the room. The tools of localization are similar to detection, but vary in processing due to the number of receiving nodes used (see Section 4.2.2).

5.2.1 Offline Data Collection and Mapping for Three Nodes.

The first step in evaluating cell-based localization was to analyze the calibration data. As mentioned in Section 3.4.2, the area of interest was divided into 25 cells, each measuring 2 ft x 2 ft (0.61 m x 0.61 m). Figure 11 in that section presented this division as well as the placement of the devices.

Despite the changes in this room from the detection setup, the silence period should still have the similar $\overline{R}_s$. To verify this, the silence period was recorded and analyzed again for five trials of 60 seconds each. As expected, $\overline{R}_s$ across all three nodes was well within the values observed during detection. To improve results and reduce calibration time, the filter threshold for both event and localization collections was manually set for all nodes. A concern from the detection experiments was that the filter threshold was too

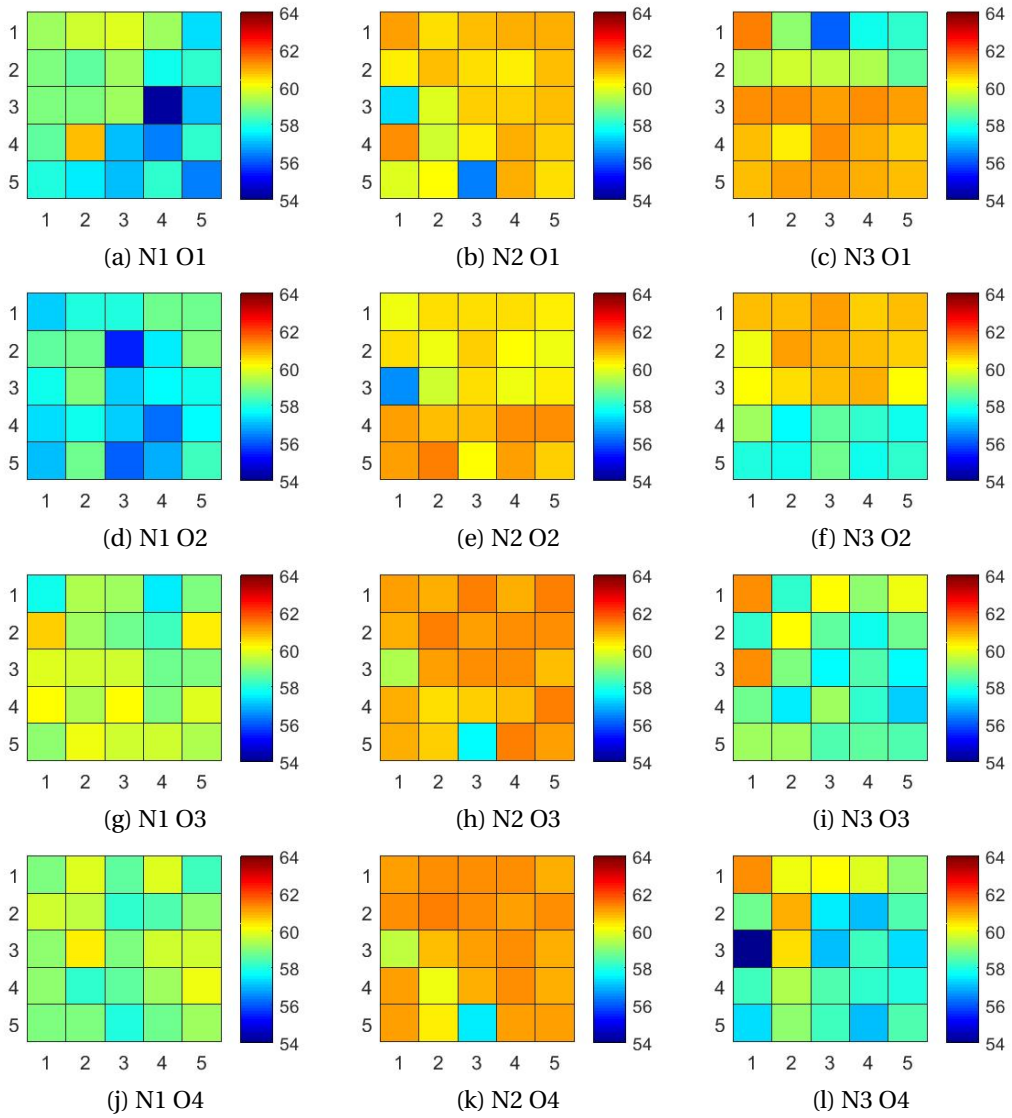
high to collect highly-attenuated RSS. Unlike simple-setup detection, fingerprinting-based localization can leverage high fluctuations to make its offline data map. Therefore, the manually-set filter threshold was lowered to one standard deviation above the midpoint of the mean and the max R_s:

$$\text{filter threshold dB} = \frac{14.0 + 34.6}{2} + 1.6 = 25.9 \tag{31}$$

which should be low enough to capture more highly-attenuated signals while still reducing the silent period.

Calibration of the target in each cell, including the empty case ($c = 0$) when the target was not in the room, was then performed. All three nodes participated in the calibration of each cell. A factor of interest in the experiment was the orientation of the target, so four different target orientations were calibrated for each cell (see Figure 12 in Section 3.4.2). As a result, a total of 104 collection periods were recorded. Each collection period was 60 seconds, during which all nodes were collecting and sending data to the DCPS laptop. In total, calibration for a 25-cell room took approximately two hours.

The $\overline{R}_e$ calibration results for all 25 cells from each of the three nodes for each of the four orientations are displayed for comparison in the heat maps of Figure 23. For certain cells, target orientation did matter and produced noticeably different $\overline{R}_e$. This likely occurred because the target would block the LOS between the Hub and the node. For example, N2 was able to distinguish the target location in cells C11 and C23 across all orientations, but O1 produced a more distinguishable C23, while O2 produced the most distinguishable C11. Similarly, N3 was able to distinguish C11, but mostly when the target was in O4. N1 should have similar behavior for O1, but in all cells and orientations from N1, there were no clear explanations for the observed $\overline{R}_e$. For N3, C1 appeared lowly attenuated in all orientations and may be important in localization. Through just a qualitative examination of these heat maps, orientations may affect performance.



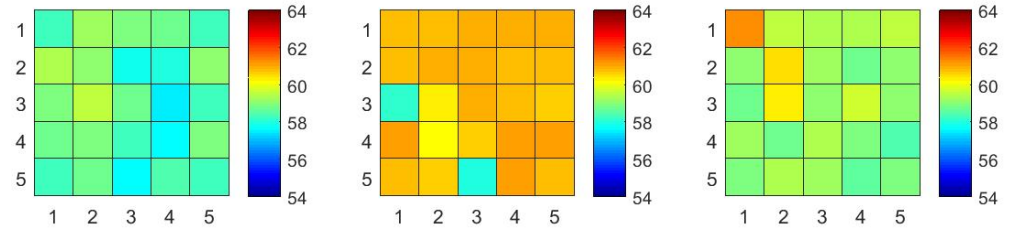
(a) N1 O1 (b) N2 O1 (c) N3 O1

(d) N1 O2 (e) N2 O2 (f) N3 O2

(g) N1 O3 (h) N2 O3 (i) N3 O3

(j) N1 O4 (k) N2 O4 (l) N3 O4

Figure 23. Calibration maps of $\overline{R}_e$ values for each node (N#) and orientation (O#) where every cell corresponds to the same cell in the Room A cell division (Fig.11)



(a) Node 1 (b) Node 2 (c) Node 3

Figure 24. Calibration maps of $\overline{R}_e$ from each node for OAvg, the data set averaging results from all orientations, where every cell corresponds to the same cell in Room A (Fig.11)

Table 18. Calibration map values of the empty case from all three nodes ($\overline{R}_e(0,n)$)

Node	$n_e(0,n)$	$\overline{R}_e(0,n)$ (dB)	$\sigma_e(0,n)$ (dB)
1	494	58.3	5.7
2	514	60.8	2.0
3	482	59.9	1.3

To compare how a data set without respect to orientation performs, the results from all four orientations were averaged into another data set, "OAvg". Figure 24 displays these averaged orientation results in a heat map for each node. For N2, the distinguishable LOS calibrations for C11 and C23 were still reflected in the averaged data. For N3, the highly-attenuated C15 for O4 and C3 in O1 became indiscernible in OAvg, but the lowly-attenuated C1 for all orientations was still discernible. From a qualitative examination, using OAvg may result in heavy reliance on c_p selections of only C1, C11, and C23 compared to mixed results expected from using the orientation-based data.

Since the empty case does not depend on target orientation, its calibration from the four orientation data sets was also averaged and used in all data sets. These calibration values are presented in Table 18. The empty-case $\overline{R}_e$ from all three nodes were relatively similar despite the different distances at which the nodes were positioned from the Hub. This indicates that the path loss was only one factor for lower RSS. Another factor could be slight variations in the hardware from device to device. Regardless, these values were also similar to the non-empty cases, which suggests that false positives may be extremely high.

None of these concerns should matter in fingerprinting when the fingerprints are unique enough to accurately select the true location. However, the heat maps suggest that there may be a lot of possibilities for matching cells to target locations since not all cells have significantly unique signatures from just three nodes.

5.2.2 Cell-Based Localization with Three Nodes.

To investigate these concerns, an experiment was conducted at every cell $c \in \{0, ..., C\}$ in the room. Only one of the four orientations was tested at each cell $c \in \{1, ..., C\}$, and these orientations were determined by a random number generator. Observations at each cell occurred for six 30-second trials. In each trial, five-second intervals ($\tau = 5$ s) were used for localization. Therefore, a total of 1,560 location results were collected in this experiment. The observation data was analyzed five ways using the different data maps from calibration: O1, O2, O3, O4, and OAvg.

The confusion matrices for this experiment are shown in Tables 37-41 in Appendix B. The performance metrics for each cell location are summarized in Table 19. The $P_D(c)$ for each cell that matches the target orientation is in boldface. The result from the data that produced the highest $P_D(c)$ is italicized. In cases where the $P_D(c)$ was the same across all data sets, the result with the lowest $P_F(c)$ is italicized.

Only for seven cells ($c \in \{3, 9, 17, 20, 23, 24, 25\}$) did the highest $P_D(c)$ belong to the data that matched the target orientation. So when orientation was taken into account, only 38% of the cells using that data set performed best among all data sets. Of these cells, $P_D(c)$ was still never higher than 35%. Of all the cells in any data set, only 21 achieved $P_D(c)$ higher than 10.0%. This is most likely low because the individual cells did not produce unique enough signatures. In fact, some cells ($c \in \{4, 16, 18, 19, 22\}$) were not selected even once for the 1,560 trials. Certain cells were more preferred on average as maximum-probability selections, especially C11 and C23. Often, high $P_D(c)$ were also associated with high $P_F(c)$ (e.g. C11 for O3 and C3 for O2).

Table 20 is a summary of the average cell results of each data set for only those trials in which the target orientation matched the data set orientation. While the average false positive rates of the cell were relatively low (average $P_F(c) \leq 6.0\%$) in these trials for all four orientation data sets, so were the average true positive rates (average $P_D(c) \leq 7.1\%$).

Table 19. Localization performance by cell ($P_D(c), P_F(c)$) based on different foam block orientation calibration data sets with $P_D(c)$ for each cell that matches the target orientation in boldface and the result from the data set that produced the highest $P_D(c)$ italicized (in case of tie, the result with the lowest $P_F(c)$ is italicized)

Target		O1 Data		O2 Data		O3 Data		O4 Data		OAvg Data	
c	O	$P_D(c)$	$P_F(c)$	$P_D(c)$	$P_F(c)$	$P_D(c)$	$P_F(c)$	$P_D(c)$	$P_F(c)$	$P_D(c)$	$P_F(c)$
1	1	**0.0%**	3.2%	0.0%	6.1%	1.7%	17.6%	3.3%	23.5%	*5.0%*	34.4%
2	2	*3.3%*	2.3%	**0.0%**	2.7%	0.0%	0.4%	0.0%	0.4%	0.0%	0.7%
3	2	1.7%	0.7%	***23.3%***	14.5%	0.0%	1.8%	0.0%	3.5%	0.0%	0.1%
4	2	0.0%	0.3%	**0.0%**	3.8%	0.0%	4.7%	0.0%	0.5%	*0.0%*	0.1%
5	1	**0.0%**	0.5%	*1.7%*	3.2%	0.0%	0.9%	0.0%	0.9%	0.0%	0.1%
6	3	*15.0%*	1.9%	3.3%	2.4%	**0.0%**	1.3%	11.7%	0.1%	10.0%	1.1%
7	1	**0.0%**	0.6%	*28.3%*	14.3%	0.0%	6.9%	10.0%	10.7%	3.3%	12.4%
8	4	0.0%	0.8%	*8.3%*	6.7%	0.0%	0.7%	**0.0%**	0.1%	0.0%	0.3%
9	1	***1.7%***	1.7%	0.0%	2.7%	0.0%	0.4%	0.0%	0.1%	0.0%	0.2%
10	3	1.7%	1.3%	*3.3%*	5.3%	**0.0%**	0.3%	0.0%	0.1%	1.7%	0.1%
11	4	15.0%	2.9%	5.0%	3.3%	25.0%	34.3%	**0.0%**	10.8%	*30.0%*	2.6%
12	3	15.0%	11.9%	*20.0%*	14.7%	**0.0%**	0.9%	11.7%	18.2%	18.3%	20.3%
13	4	*15.0%*	5.3%	3.3%	1.6%	0.0%	0.1%	**0.0%**	0.0%	0.0%	0.1%
14	3	*11.7%*	4.9%	6.7%	3.6%	**0.0%**	0.6%	0.0%	0.1%	1.7%	2.5%
15	1	**1.7%**	1.5%	1.7%	3.7%	*3.3%*	1.8%	0.0%	0.0%	0.0%	0.4%
16	1	**0.0%**	3.2%	*5.0%*	1.5%	0.0%	0.5%	0.0%	0.0%	1.7%	0.5%
17	4	10.0%	11.7%	0.0%	0.1%	3.3%	9.4%	***16.7%***	13.9%	11.7%	13.0%
18	3	0.0%	6.9%	0.0%	0.7%	**0.0%**	6.4%	*0.0%*	0.3%	0.0%	1.2%
19	1	**0.0%**	2.2%	0.0%	0.4%	1.7%	0.7%	*0.0%*	0.1%	0.0%	0.9%
20	1	***13.3%***	6.5%	0.0%	0.1%	1.7%	0.1%	0.0%	0.7%	1.7%	0.7%
21	4	*23.3%*	10.7%	0.0%	0.1%	0.0%	0.9%	**0.0%**	0.0%	0.0%	0.2%
22	4	0.0%	7.7%	*0.0%*	0.3%	0.0%	2.1%	**0.0%**	11.3%	0.0%	1.7%
23	3	5.0%	1.7%	1.7%	4.3%	***35.0%***	2.1%	18.3%	2.3%	18.3%	3.0%
24	2	1.7%	1.7%	***5.0%***	0.7%	0.0%	0.3%	0.0%	0.1%	0.0%	0.8%
25	1	***3.3%***	4.3%	0.0%	1.1%	0.0%	0.1%	0.0%	0.0%	0.0%	0.3%
0	N/A	1.7%	1.5%	0.0%	1.7%	*11.7%*	5.2%	1.7%	3.5%	3.3%	2.5%

Additionally, when taking into account all trials using the OAvg data set, similar average performance was observed. Therefore, it cannot be said that isolating the orientation of the target necessarily improved performance of this method.

Table 21 shows overall performance and accuracy for different data sets, without respect to target orientation. The overall false positives were high ($P_F \geq 88.3\%$) whereas the overall true positives and the accuracy were low ($P_D, P_A \leq 5.5\%$). High P_F were expected due to the calibrated empty-case values not being unique enough to distinguish

Table 20. Average localization performance of cells (avg $P_D(c)$, avg $P_F(c)$) for different data sets for trials where the target orientation matched the calibration orientation

Data Set	avg $P_D(c)$	avg $P_F(c)$
O1	2.2%	2.7%
O2	7.1%	5.4%
O3	5.8%	1.9%
O4	2.8%	6.0%
OAvg	4.1%	3.8%

Table 21. Accuracy (P_A) and overall localization performance (P_D, P_F) from different target orientation calibration data sets

Data Set	D (P_D)	F (P_F)	A (P_A)
O1	83 (5.5%)	59 (98.3%)	84 (5.4%)
O2	70 (4.7%)	60 (100.0%)	70 (4.5%)
O3	43 (2.9%)	53 (88.3%)	50 (3.2%)
O4	43 (2.9%)	59 (98.3%)	44 (2.8%)
OAvg	64 (4.1%)	58 (96.7%)	66 (4.1%)

the case from most of the other cells.

In general, the system's cell-based method did not perform well, even when factoring in orientation. Therefore, orientation-based matching can be unreliable and may not be better than disregarding orientation. Using the averaged data, however, had the benefits of less calibration time and less computational overhead during localization. This is important in potential deployed scenarios where the human to be localized may have any orientation in the room.

5.2.3 Remarks about the Method.

The performance of the proof of concept's cell-based localization method was highly unfavorable. Accuracy of the O1, O2, O3, O4, and OAvg data sets (P_A = 5.4%, 4.5%, 3.2%, 2.8%, and 4.1%, respectively) was only better than random guessing (1/26 = 3.8%) for three of the data sets, yet only barely so. Even orientation-based fingerprinting did not result in favorable results.

One suggestion would be to allocate even more time for calibration and investigate whether this improves the uniqueness of the signatures for localization. Another suggestion would be to change the time allocated for producing a localization result, but to do so would restrict the system's ability to locate targets that do not remain stationary during that time period. Instead, this section looks to improve the system's fingerprinting method.

Thresholds for Reducing False Positives.

One approach for reducing the probability of false positives is to implement a threshold for the probabilities $P(c,n)$ before a cell can be selected as a probable target location. Since the method uses a Welch's t-test to test how well the observation data matches the offline data, requiring that at least one p-value for a candidate cell return a minimum value of 0.05 (a normal threshold for hypothesis testing) should reduce incorrect selections. In the case that this threshold requirement is not reached by g nodes of any of the candidate cells, then the empty case would be the reported result ($c_p = 0$).

For example, let g be the minimum number of nodes in a cell for which the p-value must be greater than 0.05, and for some trial with $N = 3$ and just two possible cell locations, C1 and C2 (and the empty case), let the true location be C2. If the p-values for C1 were $P(1,1) = 0.00002$, $P(1,2) = 0.64$, and $P(1.3) = 0.04$ (0.23 average) and for C2 were $P(2,1) = 0.24$, $P(2,2) = 0.37$, and $P(2,3) = 0.03$ (0.21 average), then the current method would choose C1 ($c_p = 1$) because of the higher average. With the added threshold requirement, if $g = 0$ or 1, a false location would be reported. But if $g > 1$, C1 would no longer be a location candidate for c_p because two of its values were less than 0.05. C2 would then be correctly identified as the target location. However, if $g = 3$, the empty case would be reported because $P(2,3) > 0.05$ and so $g = 3$ was not satisfied.

Table 22 shows the accuracy and overall performance for each data set after this

Table 22. Accuracy (P_A) and overall localization performance (P_D, P_F) for cell-based localization using threshold requirements (g)

	$g=0$			$g=1$			$g=2$			$g=3$		
Data Set	P_D	P_F	P_A	P_D	P_F	P_A	P_D	P_F	P_A	P_D	P_F	P_A
O1	5.5%	98.3%	5.4%	5.5%	98.3%	5.4%	5.7%	98.3%	5.5%	5.4%	66.7%	6.5%
O2	4.7%	100.0%	4.5%	4.7%	100.0%	4.5%	4.4%	98.3%	4.3%	4.6%	58.3%	6.0%
O3	2.9%	88.3%	3.2%	2.9%	88.3%	3.2%	2.7%	78.3%	3.4%	0.7%	36.7%	3.1%
O4	2.9%	98.3%	2.8%	2.9%	98.3%	2.8%	2.6%	88.3%	3.0%	0.8%	21.7%	3.8%
OAvg	4.1%	96.7%	4.1%	4.1%	96.7%	4.1%	3.9%	88.3%	4.2%	0.9%	28.3%	3.6%
Average	4.0%	96.3%	4.0%	4.0%	96.3%	4.0%	3.9%	90.3%	4.1%	2.5%	42.3%	4.6%

method update was applied to the same data from above. Figure 25 is the accompanying graph displaying ROC for each data set and shows that no data set achieved P_D and P_F performance better than random guessing, but as g increased, performance also improved. The changes from $g = 0$ to $g = 1$ were nonexistent. When $g = 2$, slight changes were observed. Interestingly, P_D of the O1 data set increased from 5.5% to 5.7%. All other data sets produced small decreases in P_D and a decrease in P_F of up to 10.0%. On average, P_A increased slightly from 4.0% to 4.1%.

Applying this threshold requirement on all nodes (when $g = 3$) produced greater changes. The average P_F for all data sets was only 42.3% compared to 96.3% in the original method. P_F of a data set decreased to as low as 21.7% from values greater than 88.3%. The lowest decrease in any one data set from the original method was by 31.6%, but the greatest was by 76.6%. However, P_D also decreased for all data sets. The change in P_D was small (0.1%) for O1 and O2 data sets, but was more significant in the other data sets. For example, for OAvg, P_D went from 4.1% to 0.9%. However, P_A increased for the O1, O2, and O4 data sets, producing an average P_A of 4.6% from all data sets.

These results demonstrated that increasing g could decrease P_F but often also decreased P_D. This means that X0 was selected more as g increased. This suggests that many locations had similar average p-values but also relied on high results from just one or two nodes, so inaccurate locations were reported. Thus, g is useful for reducing P_F by increasing the likelihood that $x_p = 0$ as opposed to reporting inaccurate locations.

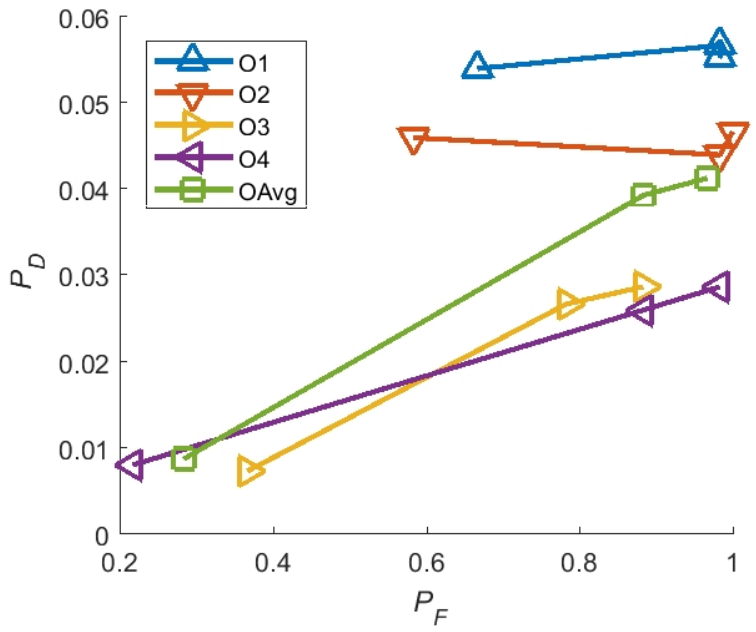


Figure 25. ROC graph of overall performance (P_D, P_F) with varying g for each orientation data set for cell-based localization (left to right is decreasing g)

Proximity for Increased Accuracy.

Minimal improvements in P_A were observed from implementing g above. A large reason for these poor results was that the calibration data for all 25 cells and the empty case were not unique enough to select the cell in which a target was actually located. Therefore, instead of continuing with the cell-based method, this section looks also towards an alternate approach for improving P_A.

Considering the application of home automation devices, a DfP system does not necessarily need to be able to locate targets throughout an entire room. In fact, home automation networks are deployed typically to allow remote access of common appliances. Therefore, an exhaustive calibration map like that of the cell-based method may be unnecessary. Home automation applications are not confined to simple area like cells, but rather expand to an area defined by context and situation. For example, a daily alarm could be configured to not go off if the user is not located on the bed. Other ar-

eas in the room could be not interesting for localization. This concept is context-based fingerprinting.

To transition from cell-based methods and argue in favor of context-based methods, a proof of concept that a similar method would produce better P_A can be shown by adapting the previous experiment to allow proximity selections as a true positive. In the experiment above, the room was a 5 x 5 grid of cells, and localization was performed such that only one of those cells could be the true result. However, by expanding localization to larger areas, P_A could greatly improve. For example, instead of just a single cell as a returned result, let adjacent cells also be true results. Then localization of a target in C1 ($c_{true} = 1$), for example, would be positive if C1, C2, C6, or C7 were returned ($c_p \in \{1,2,6,7\}$).

Thus, this section investigates P_A changes due to proximity selection including adjacent cells as opposed to strict selection of single cells. The results could either only remain the same or improve, but improvements would suggest that context-based fingerprinting would produce better localization P_A and that using strict cell approximations could be too restrictive in this system. The same calibration and observation data as from above was used. During analysis, all adjacent cells and the actual cell where the target is located were considered positive matches. The P_A results of this proximity-based system are listed in Table 23 and shown in Figure 26. Both also present the previous results from strict-cell fingerprinting for comparison.

The five data sets went from average $P_A = 4.0\%$ to 22.2% with the proximity-based approach. This average improvement suggests that many cells in the offline data maps did not produce unique enough signatures, especially in comparison to their adjacent cells. Therefore, further reducing the set of possible locations of a target and limiting these locations to non-adjacent regions could improve the localization method for deployment on top of home automation devices.

Table 23. Comparison of accuracy (P_A) for strict cell and proximity methods by data set

Data Set	Strict Cell P_A	Proximity P_A
O1	5.4%	31.3%
O2	4.5%	25.1%
O3	3.2%	20.8%
O4	2.8%	17.1%
OAvg	4.1%	16.8%
Average	4.0%	22.2 %

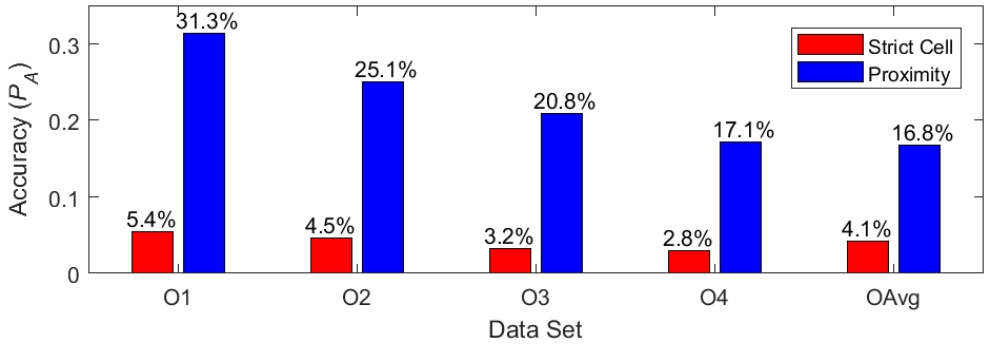


Figure 26. Comparison of accuracy (P_A) for strict cell and proximity methods by data set

The cell-based fingerprinting method no longer appears plausible for the current proof-of-concept system due to its poor performance in P_D, P_F, & P_A. But by expanding to a context-based fingerprinting method using threshold requirements, both P_A and P_D could improve and P_F could decrease. This was suggested by improved results after implementing a proximity-based approach and a threshold parameter in analysis of the cell-based localization experiment. However, a true context-based system should also reduce costs from calibration. This includes reducing data that needs to be collected, processed, and stored as well as reducing the nearly two hours of calibration for a 10 ft x 10 ft room.

5.3 Localization with a Context-Based Method

The previous localization experiment was met with unfavorable results. As a result, two considerations to the method were investigated in Section 5.2.3 to improve upon the low P_A, high P_F, and high calibration costs of the cell-based fingerprinting method. This section outlines the context-based, maximum-probability fingerprinting method that is the result of those efforts. This section also tests the method's performance by conducting an experiment for locating a target in only a few regions of interest.

The cell-based method was ambitious in its desire to locate a target in all possible cells in a room. However, since home automation is an intended use of this proof-of-concept system, an accurate response at all of these cells would be excessive. For example, localization of a target in a corner of the room that no one frequents is unnecessary. Thus, while context-based methods are limited in precision, it focuses only on localization of a target in regions that could provide useful information.

Many uses of home automation networks are based on the principle of being able to access and control appliances and settings remotely. High precision is not necessary for many uses of these devices, so localization in a single room can instead be separated into just a few context regions. One example of such an application is keeping the living room lights on for as long as the user occupies the TV couch. A smart personal assistant can prompt the user with available food choices just by localizing the user in the vicinity of the kitchen near meal times. Music could move around the house with the user seamlessly, without the current limitation of having to actively prompt devices to broadcast elsewhere. Other uses of home automation are similar; they are responses to human needs, and context-based localization can be used to assess these needs.

5.3.1 Methodology.

This method is still a maximum-probability fingerprinting method, so most of the principles from the previous method still apply. For example, the hardware does not change. The response variable is still the observed mean RSS in dB calculated using (11) with the same MATLAB® command seen in Section 4.2.2. The main control variable for all experiments is target location. The Welch's t-test described in Definition 4 is still used to compare the observed means, $\overline{R}(t_1, n)$, over a period of time, t_1, from each node, n, to the offline data matrix values for each possible context region.

The main changes are those that incorporate the considerations investigated in Section 5.2.3: reducing P_F by adding a threshold for the maximum average p-values and improving P_A by decreasing the set of possible locations while still covering a large area. Thus, this methodology amends that described in Section 3.3.2.

The following assumptions for this context-based method are added to those in Section 3.1 to further improve performance:

Assumption 7 *A context region provides adequate localization precision.*

Assumption 8 *Each context area is chosen such that it should produce a unique signature from the other context areas. Specifically, no areas should overlap.*

Assumption 9 *No more than five context regions should exist in a room, and this number should be lower for smaller rooms.*

Assumption 10 *There are at least three nodes in the system.*

Instead of dividing the entire area of interest and collecting offline data for each of the cells, data is collected for only the contexts of interest. A context region is described as $x \in \{1, 2, ..., X\}$, where $X \leq 5$ in any one room. Each context region should not overlap

with other regions to more likely produce a unique RSS signature. There are at least three nodes in the system ($N \geq 3$) to increase reliability.

During offline calibration, only the x regions need to be calibrated. All other regions are designated to be the empty case ($x = 0$), when the target is either not in the room or not in a region of interest. The empty case is not calibrated. Since this method only has X cases compared to $C + 1$ cases, total calibration costs are significantly reduced.

Calibration collections are still collected over a period T from all N nodes in the room when the target is in some $x \in \{1, 2, ...X\}$. However, instead of remaining stationary during calibration, the target is free to roam or sweep the region of interest. This is to more accurately depict the entire region as opposed to just a cell. Furthermore, since target orientation was not determined to be significant in affecting system performance as compared to an average of the orientations, the target is also allowed to assume any orientation during calibration. The result from one node n for the region x is some mean RSS, $\overline{R}_e(x, n)$, with standard deviation, $\sigma_e(x, n)$, from some $n_e(x, n)$ collection steps.

During the online observation phase, data is collected similarly to before: all N nodes simultaneously collect RSS data starting at time t_0. Each node n computes and stores the mean RSS over a time period, $\tau < T$, in $\overline{R}(t_1, n)$ where time $t_1 = t_0 + m \cdot \tau$ for some positive integer m. The accumulation of $\overline{R}(t_1, n)$ for all nodes is $\overline{R}(t_1)$, a vector of size N. Then at some time t_1 during which human presence occurs, the context region, x_{true}, in which human presence exists is attempted to be matched by the most probable region, x_p, by comparing the observed data vector $\overline{R}(t_1)$ to each of the calibrated $\overline{R}_e(x, n)$ and finding the region for which the maximum probability of match occurs.

Again, the probability of match for a sample at some t_1 is defined by the average p-values calculated using two-sided Welch's t-tests comparing the observed $\overline{R}(t_1, n)$ to the offline $\overline{R}_e(x, n)$ for each node, as described in Definition 4. However, there is an additional argument for selecting a region that involves the minimum number of $P(x, n)$

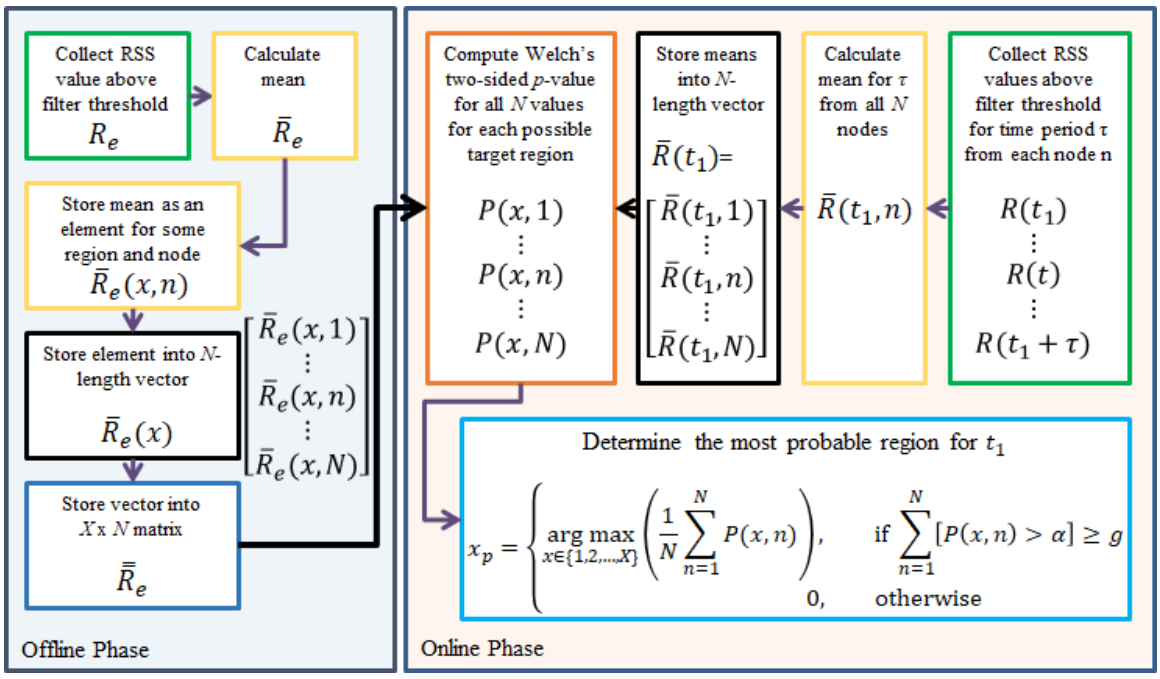


Figure 27. Flow chart depicting the context-based localization method

values, g, for a candidate region, x, that must have p-values greater than the significance level, α.

Definition 7 *Let $P(x,n)$ be the probability that the observed value is close to the offline value for some context region, x, from some node, n. Then the selected location region, x_p, at t_1 is some x for which the probability of match for a sample at some time t_1 is the maximum AND for which at least g values of $P(x,n) > \alpha$ for x among all $n \in \{1,\ldots,N\}$ nodes, where $\alpha = 0.05$ and g is experimentally varied. If there is no x for which this argument is satisfied, then the selected context region is the empty case, $x = 0$; i.e.,*

$$x_p = \begin{cases} \arg\max_{x\in\{1,2,\ldots,X\}} \left(\frac{1}{N}\sum_{n=1}^{N} P(x,n)\right) & \text{if} \sum_{n=1}^{N}[P(x,n)>\alpha] \geq g \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

Figure 27 depicts this method without the examples of Figure 9.

5.3.2 Evaluating Context-Based Localization.

To evaluate this new method, localization was performed again. As previously mentioned, the response variables were RSS values and the control variable was the target location in a region. An additional control parameter was g. The same evaluation metrics used in cell-based localization were used here in context-based localization, except the parameter was now regions, x, not cells, c (see Section 3.3.2). These metrics are P_A, P_D, P_F, $P_D(x)$, and $P_F(x)$.

Experiment Setup.

Room A was used as described in Section 3.4.2, but with five nodes and only three possible locations. The first location, X1, was the light switch that controls the ceiling light. The second location, X2, was the closet door. A possible application for localizing the target would be to prompt a smart assistant for the weather to help the target decide what to wear. The third location, X3, was the window. A possible application would be to change the angle of the blinds to keep the sun from harming the target's eyes. Figure 28 depicts this new setup.

Offline Data Collection and Mapping.

For simplicity, the manual silence period filter threshold of 25.9 dB was used again. The time period for more nodes using the proof of concept system required a longer collection period because collection was not parallel. Therefore, calibration of the three locations was five minutes each. The script was paused every fifteen seconds to change the orientation of the foam blocks and their specific location within the region. The results of each region's $\overline{R}_e(x,n)$, $\sigma_e(x,n)$, and $n_e(x,n)$ from each node are listed in Table 24.

An important observation from these calibrations was that the nodes positioned in or near a region such that the node was likely to be blocked were more likely to observe

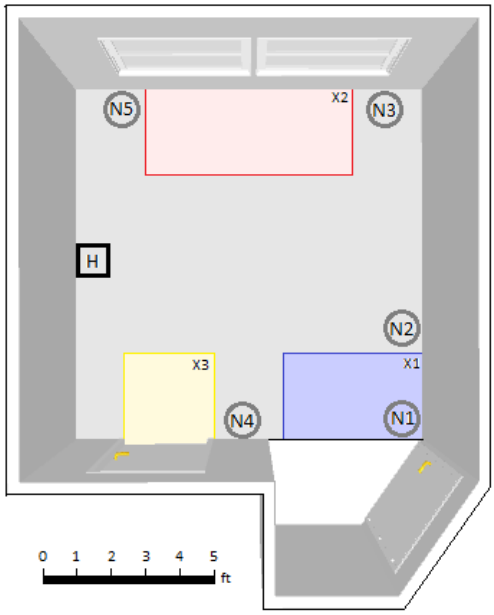


Figure 28. Diagram of Room A showing regions (X#) for context-based localization

Table 24. Summary statistics from context calibration of Room A

		Node														
		N1			N2			N3			N4			N5		
		n_e	$\overline{R}_e$	σ_e	n_e	$\overline{R}_e$	σ_e	n_e	$\overline{R}_e$	σ_e	n_e	$\overline{R}_e$	σ_e	n_e	$\overline{R}_e$	σ_e
Region	X1	351	57.1	8.7	331	61.3	1.8	328	61.4	1.7	330	59.0	4.7	373	52.6	10.8
	X2	355	57.8	6.3	357	61.2	1.6	385	61.1	2.6	357	58.4	5.1	350	52.8	10.4
	X3	337	58.7	5.5	297	61.3	1.7	311	60.3	1.8	369	56.4	7.8	379	52.9	10.3

higher fluctuations. For example, X1-N1 saw a smaller $\overline{R}_e$ and a larger σ_e compared to X2-N1 or X3-N1. X3-N4 was the same. X2-N3 also observed higher fluctuations than did X1-N3 or X3-N3 due to the node's position to the right of X2; the calibrated target blocked the LOS only briefly near the edge of the region. These attributes should help uniquely define the regions in the offline data map.

Other nodes not near a region of interest also showed interesting results: N5 for instance had very low $\overline{R}_e$ and high σ_e in all cases. However, these were relatively the same, which is consistent with the fact that the node was never blocked during calibration. N2 saw a similar behavior for all three regions, except $\overline{R}_e$ and σ_e were more typical.

The offline data results from calibration produced noticeable behaviors from different nodes for different regions, which suggested that localization would be more accurate than cell-based localization. Limiting the number of locations allowed for this. Also, with this method, only fifteen minutes were needed for calibration. This was a significant decrease from the two hours of the cell-based method, which also only allocated 60 seconds (4 min for averaged orientation) for each cell.

Observation Phase and Analysis Results.

One trial was performed at each of twenty different locations. Each trial was 60 seconds long, and $\tau = 20$ s for selecting a most probable region (an increase from 5 s because of the increase in nodes for a non-parallel collection script). As a result, three location results were reported for every trial, for a total of 60 results for the entire experiment. The results from each trial are listed in Table 42 in Appendix B. In this section, Table 25 presents the confusion matrices with varying g.

The confusion matrix shows that the threshold parameter did not affect the results until $g > 2$, which was expected based on observations of the calibration data. As g increased, selecting the empty case ($x_p = 0$) was more prevalent (from just 2 selections when $g \leq 2$ to 38 when $g = 5$). The X1 region was often confused for a target in the empty case. This is likely because of how a target in X1 would only greatly affect N1. X1 was also the most prevalent region selected among the three regions and was reported up to 30 times (50%). This false predictor was reduced with increasing g, but suggests that data did not fluctuate enough among different regions in the empty apartment room to produce perfect results.

The performance metrics of the individual regions are presented in Table 26. These results are also displayed in Figure 29 to examine the effect of g on region performance and also in the ROC graph in Figure 30. When $g \leq 2$, X1 and X3 were both highly success-

Table 25. Confusion matrices with varying g for context-based localization in Room A

		Actual Region				
		X1	X2	X3	X0	
Predicted	X1	9	10	1	9	29
	X2	4	3	4	3	14
	X3	1	2	9	3	15
	X0	1	0	1	0	2

(a) $g = 0$

		Actual Region				
		X1	X2	X3	X0	
Predicted	X1	9	10	1	9	29
	X2	4	3	4	3	14
	X3	1	2	9	3	15
	X0	1	0	1	0	2

(b) $g = 1$

		Actual Region				
		X1	X2	X3	X0	
Predicted	X1	9	10	1	9	29
	X2	4	3	4	3	14
	X3	1	2	9	3	15
	X0	1	0	1	0	2

(c) $g = 2$

		Actual Region				
		X1	X2	X3	X0	
Predicted	X1	9	11	1	9	30
	X2	4	2	4	3	13
	X3	1	2	8	2	13
	X0	1	0	2	1	4

(d) $g = 3$

		Actual Region				
		X1	X2	X3	X0	
Predicted	X1	5	9	2	7	23
	X2	5	3	1	3	12
	X3	1	2	6	1	10
	X0	4	1	6	4	15

(e) $g = 4$

		Actual Region				
		X1	X2	X3	X0	
Predicted	X1	4	4	0	6	14
	X2	3	1	1	0	5
	X3	0	0	3	0	3
	X0	8	10	11	9	38

(f) $g = 5$

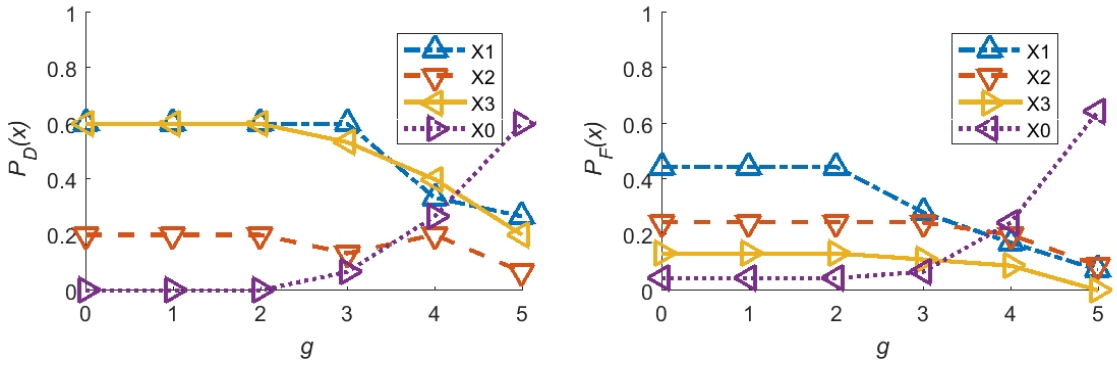
ful with $P_D(1), P_D(3) = 60.0\%$. X2 was also able to achieve $P_D(2) = 20.0\%$. However, X0 was not correctly identified once. P_F was also much better than in the cell-based case, but X1 still had $P_F(1) = 44.4\%$.

As seen in the confusion matrix, as g increased, the average $P_F(x)$ should have decreased as more empty-case selections were reported. This was observed for $g = 3$ and 4 with an improvement from average $P_F(x) = 21.7\%$ to 17.6%. This was mostly attributed to the lower $P_F(1)$, which was 28.0% for $g = 3$ and 17.1% for $g = 4$, as a result of more frequent $x_p = 0$. $P_F(x)$ of other regions also decreased, but that of X0 increased greatly, up to 64.4% when $g = 5$. Nonetheless, average $P_D(x)$ in all g was much better compared to average $P_D(c)$ of the cell-based method.

To better depict this method's performance, Table 27 lists the accuracy (P_A) and overall performance metrics (P_D, P_F) with varying g. (The next section graphs these metrics and compares them to those for two other scenarios). P_A using this method was better than the 25.0% probability of random guessing for four regions. Higher g lowered P_A. The best accuracy was $P_A = 35.0\%$ when $g \leq 2$, but only decreased to 33.3% at $g = 3$, 30.0% at $g = 4$, and 28.3% at $g = 5$.

Table 26. Performance by region $(P_D(x), P_F(x))$ for context-based localization using threshold requirements (g) in Room A

	$g=0$		$g=1$		$g=2$		$g=3$		$g=4$		$g=5$	
Region	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$
X1	60.0%	44.4%	60.0%	44.4%	60.0%	44.4%	60.0%	28.0%	33.3%	17.1%	26.7%	7.4%
X2	20.0%	24.4%	20.0%	24.4%	20.0%	24.4%	13.3%	24.4%	20.0%	20.0%	6.7%	8.9%
X3	60.0%	13.3%	60.0%	13.3%	60.0%	13.3%	53.3%	11.1%	40.0%	8.9%	20.0%	0.0%
X0	0.0%	4.4%	0.0%	4.4%	0.0%	4.4%	6.7%	6.7%	26.7%	24.4%	60.0%	64.4%
Average	35.0%	21.7%	35.0%	21.7%	35.0%	21.7%	33.3%	17.6%	30.0%	17.6%	28.3%	20.2%



(a) $P_D(x)$ (b) $P_F(x)$

Figure 29. Effect of varying g on region performance $(P_D(x), P_F(x))$ in Room A

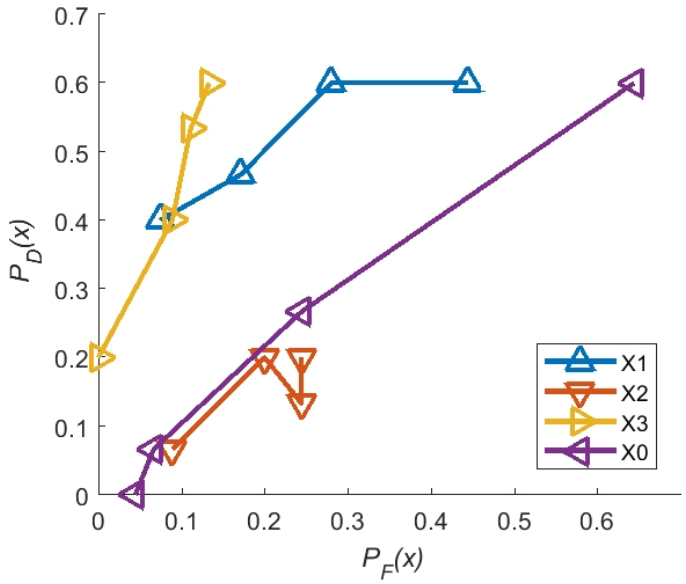


Figure 30. ROC graph for context-based region performance $(P_D(x), P_F(x))$ with varying g in Room A (for X1-3, left to right is decreasing g; for X0, left to right is increasing g; dashed line is random guess reference)

The trend of P_D with increasing g was similar; detection was highest at 46.7% when $g \leq 2$ and decreased slightly to 42.2% at $g = 3$, but drastically lowered to 31.1% at $g = 4$ and then 17.8% at $g = 5$. Based on P_A and P_D alone, enforcing $g = N$ may be too strict and may cause fewer correct selections than desired for a system.

However, P_F favorably decreased as a result. It was at 100.0% until $g = 3$, where it decreased slightly to 93.3%, then more drastically to 73.3% at $g = 4$ and only 40.0% at $g = 5$. In many cases, a lower P_F is more acceptable than a high P_A or P_D. Therefore, in this scenario, it appeared that the most acceptable performance was when $g = 5$. Regardless, this system performed much better with context regions than with cells.

Summary.

A context-based method was developed in this section by adapting the cell-based method to improve P_A and lower P_F and calibration time. The changes were a threshold parameter, g, for the number of nodes in region selection that must return a p-value greater than 0.05 and a limit, $X < 5$, to the set of possible locations to just a few context regions. This would optimize the system to give users target location only in regions of interest instead of a whole room. With fewer possible locations, calibration time was also reduced or could be reallocated to producing better calibration maps.

The experiment conducted using this method produced the expected improvements. With just three regions of interest (four regions to select from, including the empty case), the system was able to achieve accuracy $28.3\% \leq P_A \leq 35.0\%$. The cell-based method only achieved $P_A \leq 5.38\%$. Detection using this new method was also higher with $17.8\% \leq P_D \leq 46.7\%$ compared to the cell-based $P_D \leq 5.53\%$. Also, the cell-based method observed a P_F only as low as 88.3%, but this method was able to reduce P_F down from a worst-case 100.0% to 40.0%.

The context-based method then is an improvement upon the cell-based method.

Table 27. Accuracy (P_D) and overall localization performance (P_F) for context-based localization using threshold requirements (g) in Room A

g	P_D	P_F	P_A
0	46.7%	100.0%	35.0%
1	46.7%	100.0%	35.0%
2	46.7%	100.0%	35.0%
3	42.2%	93.3%	33.3%
4	31.1%	73.3%	30.0%
5	17.8%	40.0%	28.3%

Though the results are not optimal, they are promising for future development of this DfP localization system using home automation devices.

5.4 Investigating System Robustness in Different Rooms

The previous section conducted localization an empty room. Despite not achieving perfect accuracy, the proof of concept could still benefit from deploying in scenarios that are more realistic and evaluating the performance changes. These scenarios were introduced in Section 3.4.3 with the hope of using cell-based localization. However, as the previous section demonstrated, a context-based approach produces better results. Therefore, this section presents experiments that investigated localization results in other environments.

5.4.1 Localization in Room B.

The second scenario allowed an investigation into how the system performs in a room that is not empty and that has only two context regions. Room B is shown in Figure 31 with these two regions. X1 was the space near the computer desk. Potential applications would be to monitor the time the user spends at the computer or to activate the desktop monitor as the user approaches the desk. X2 was the space near the closet.

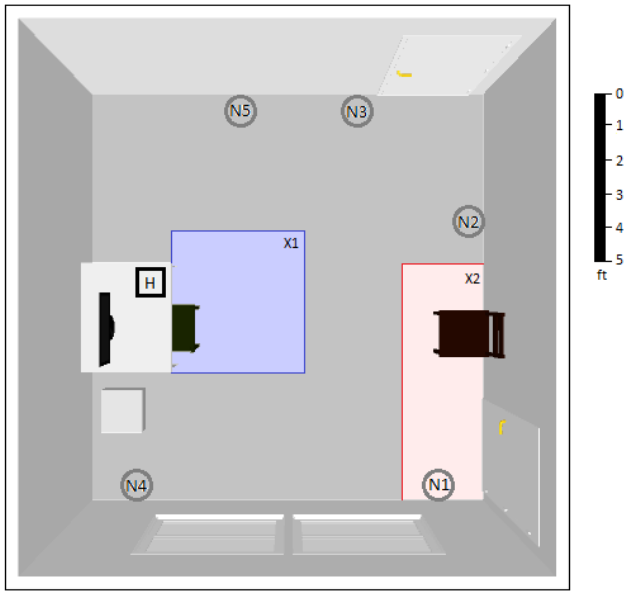


Figure 31. Diagram of Room B showing regions (X#) for context-based localization

Table 28. Summary statistics from context calibration of Room B

		Node														
		N1			N2			N3			N4			N5		
		n_e	$\overline{R}_e$	σ_e	n_e	$\overline{R}_e$	σ_e	n_e	$\overline{R}_e$	σ_e	n_e	$\overline{R}_e$	σ_e	n_e	$\overline{R}_e$	σ_e
Region	X1	521	47.2	6.6	551	59.7	5.5	539	60.0	3.3	532	60.8	2.1	542	60.3	3.3
	X2	619	46.4	6.7	636	60.8	2.7	614	60.3	2.4	627	60.2	2.8	629	60.4	2.7

All nodes except N4 should be affected by X1 since the target should block the LOS of these nodes some time during calibration. The X2 region may not be well distinguished with larger g because the target would only affect N4 and only in part of the region. Therefore, the X2 selection may often be confused with the empty case selection. Since this scenario only had two context regions, the expected P_A performance should be at least 33.3%.

Calibration.

Since Room B only had two context regions, additional calibration time was given to both regions. Calibration was a total of fifteen minutes like in Section 5.3.2. Arbitrarily,

the division of the five extra minutes was two minutes to X1 and three to X2 for a total of seven and eight minutes, respectively. The calibration results are shown in Table 28.

The calibration of X1 and X2 did not produce data that was as different as what would have been ideal for fingerprinting. Instead, similar to the results from Room A, the changes in position and orientation of the target within a region produced high σ_e and similar $\overline{R}_e$ for all nodes. Only the behavior in one node should have been reflected during X2 calibration, but most nodes just fluctuated normally. N1 and N4 both recorded a decrease in $\overline{R}_e$ and a higher σ_e. N4's behavior could be attributed to reflections. N2's behavior was consistent with its distance and how the target would block the LOS a bit. These results however were not significantly different from X1 to X2.

Compared to X2, X1 should have decreased $\overline{R}_e$ in all but just one node. Due to the orientation and positioning of the target, σ_e should also have been high for X1. For three of these nodes, these behaviors should not have been observed for X2. N2, N3, and N5 did reflect these expected behaviors, but perhaps not significantly enough.

From a qualitative observation of these signatures, these results may not be significant enough for x_p to distinguish X1 from X2. The added concern is that the positioning of the nodes and the context regions were not optimal for prediction and that the empty case should also not be very distinguishable from X2, which would mean higher P_D and P_F when X0 is not be reported.

Observation Results.

Fifteen trials were conducted in Room B (five for each of the possible three cases). In each trial, the target was positioned and oriented randomly in the region. Collections were each one minute long, and the localization interval was $\tau = 20$ s. The total number of localization events was 45. The results from each trial are shown in Table 43 in Appendix B. The confusion matrices in Table 29 summarize these results.

Table 29. Confusion matrices with varying g for context-based localization in Room B

		Actual Region			
		X1	X2	X0	
Predicted	X1	6	4	9	19
	X2	9	11	6	26
	X0	0	0	0	0

(a) $g = 0$

		Actual Region			
		X1	X2	X0	
Predicted	X1	6	4	9	19
	X2	9	11	6	26
	X0	0	0	0	0

(b) $g = 1$

		Actual Region			
		X1	X2	X0	
Predicted	X1	6	4	9	19
	X2	9	11	6	26
	X0	0	0	0	0

(c) $g = 2$

		Actual Region			
		X1	X2	X0	
Predicted	X1	6	2	5	13
	X2	9	10	9	28
	X0	0	3	1	4

(d) $g = 3$

		Actual Region			
		X1	X2	X0	
Predicted	X1	7	2	2	11
	X2	6	7	3	16
	X0	2	6	10	18

(e) $g = 4$

		Actual Region			
		X1	X2	X0	
Predicted	X1	3	1	0	4
	X2	2	1	0	3
	X0	10	13	15	38

(f) $g = 5$

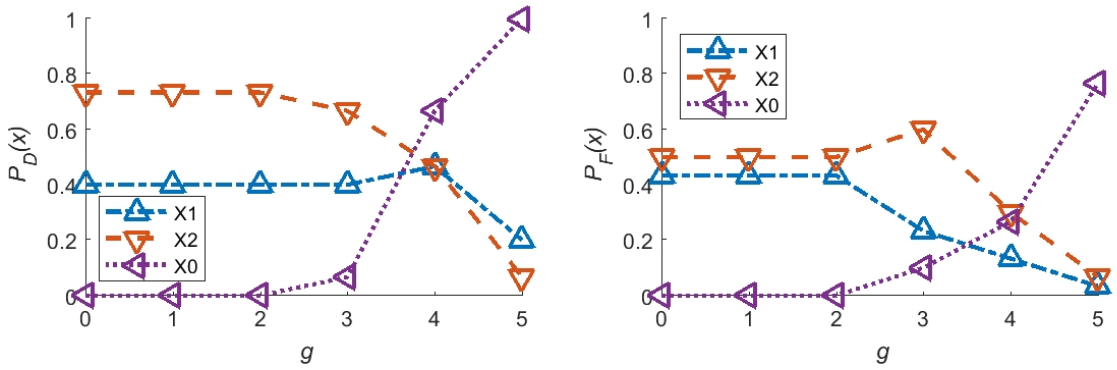
As expected, $x_p = 0$ was uncommon when $g \leq 2$. Even with $g = 3$, this empty case was only selected four times. However, when $g = 4$, $x_p = 0$ was highly prevalent at 18 times. Even more, when $g = 5$, $x_p = 0$ accounted for 38 of the possible 45 events (84.4%). This supports the calibration observation of how similar the values were in all nodes. X0 was difficult to select until $g = 5$ because X1 affected four nodes and X2 only affected one. The method preferred X1 and X2 when possible, and since the calibration of all were similar, selecting X0 was unlikely. Only when g was high could more discrimination take place such that $x_p = 0$ was more frequent.

The confusion matrices also depict how X1 and X2 were commonly confused when $g \leq 4$. However, $x_p = 2$ was more frequently reported for a target in X1. This is likely because when only enforcing g on a couple of nodes, N1 achieved high $P(2,1)$ and was sufficient for $x_p = 2$. Whereas when incorrectly localizing the target in X2 with $x_p = 1$, it was difficult to achieve larger $P(1,n)$ than $P(2,n)$ from any node.

The individual region performance metrics are presented in Table 30. Figures 32 and 33 additionally depict the performance trends from each region with varying g. On ave-

Table 30. Performance by region $(P_D(x), P_F(x))$ for context-based localization in Room B using threshold requirements (g)

	$g=0$		$g=1$		$g=2$		$g=3$		$g=4$		$g=5$	
Region	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$
X1	40.0%	43.3%	40.0%	43.3%	40.0%	43.3%	40.0%	23.3%	46.7%	13.3%	20.0%	3.3%
X2	73.3%	50.0%	73.3%	50.0%	73.3%	50.0%	66.7%	60.0%	46.7%	30.0%	6.7%	6.7%
X0	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	6.7%	10.0%	66.7%	26.7%	100.0%	76.7%
Avg	37.8%	31.1%	37.8%	31.1%	37.8%	31.1%	37.8%	31.1%	53.3%	23.3%	42.2%	28.9 %



(a) $P_D(x)$ by Region (b) $P_F(x)$ by Region

Figure 32. Effect of varying g on region performance $(P_D(x), P_F(x))$ in Room B

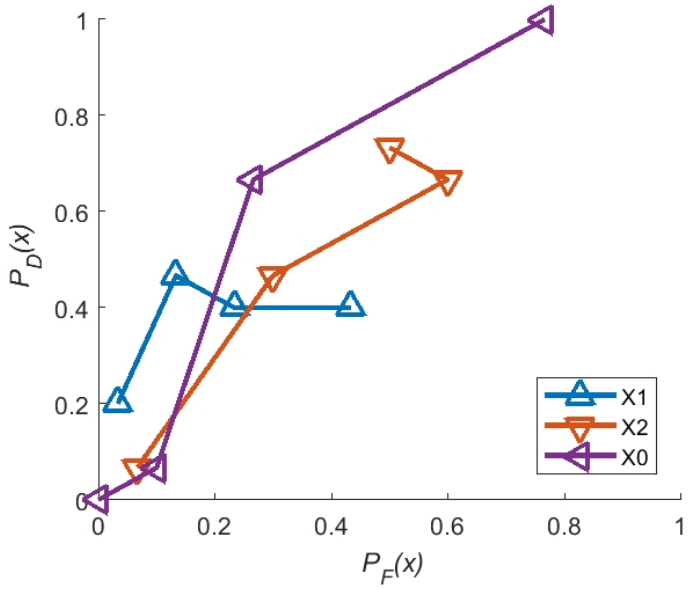


Figure 33. ROC graph for context-based region performance $(P_D(x), P_F(x))$ with varying g in Room B (for X1 & X2, left to right is decreasing g; for X0, left to right is increasing g)

rage, the regions saw true positives $37.8\% \leq P_D(x) \leq 53.3\%$ and false negatives $23.3\% \leq P_F(x) \leq 31.1\%$. Therefore, on average, the individual regions performed well. However, this was mostly because of the lack of $x_p = 0$ until g was higher. The best performance was when $g = 4$ where $x_p = 1$ and $x_p = 2$ still occurred in conjunction with $x_p = 0$ as opposed to the other cases when either $x_p = 0$ hardly ever occurred or was the dominant occurrence.

This empty case saw both ends of the $P_D(0)$ spectrum and either end was often accompanied by a similar $P_F(0)$; e.g. $(P_D(0), P_F(0)) = (0.0\%, 0.0\%)$ for $g \leq 2$, and for $g = 5$, $(P_D(0), P_F(0)) = (100.0\%, 76.7\%)$. The other region selections also saw high $P_D(x)$, but only at the cost of a high $P_F(x)$. The average performance was greatly influenced by these extremes observed for X0.

Table 31 presents the accuracy and overall performance metrics. When $g = 5$, the best P_F was observed; $P_F = 0.0\%$ because $x_p = 0$ was so frequent. P_A also increased from 37.8% when $g \leq 3$ to 53.3% when $g = 4$ and then decreased to 42.2% when $g = 5$. However, the similar P_A does not reflect a small change in successful selections for all regions. As previously shown, the similarly high P_A is mostly because in all of the X0 trials, the target was 100.0% correctly located, but only at the expense of low frequency of $x_p = 1$ and $x_p = 2$ that made it difficult for targets in either X1 or X2 to be correctly located.

Higher g in this scenario was able to produce ideal P_F but failed to more successfully locate the target in the other regions. This was because calibration did not produce unique signatures between the two regions. This could be because of non-optimal positioning of the nodes with respect to the regions. Nevertheless, P_A and P_D were still better than random guessing.

Table 31. Accuracy (P_A) and overall localization performance (P_D, P_F) for context-based localization in Room B using threshold requirements (g)

g	P_D	P_F	P_A
0	56.7%	100.0%	37.8%
1	56.7%	100.0%	37.8%
2	56.7%	100.0%	37.8%
3	56.7%	93.3%	37.8%
4	46.7%	33.3%	53.3%
5	42.2%	0.0%	42.2%

5.4.2 Localization in Room C.

The third scenario allowed an investigation of the effects of more absorbent materials in the room on localization performance. Room C was discussed in Section 3.4.3. Most important was that there was a large bed in the room as well as other absorbent materials like clothes, books, and pillows strewn about. Figure 34 depicts the localization setup of the room with larger furniture. Ideally, fingerprinting-based localization would still perform as normal despite these absorbent materials. However, due to the small differences seen in previous experiments, it was more likely that the performance would be worse, especially with nodes positioned near an absorbent object.

The three context regions were still chosen based on potential uses. X1, for example, was the bedside table, for which the method could be used to determine when the target is ready for bed or has gotten up in the morning. X2 was the area outside of the bathroom and could be used to monitor bathroom habits. X3 was once again the area near the light switch and could be used to activate the voice assistant for when the target heads out of the room, ready to start their day.

Based on the locations of the nodes with respect to the regions, the calibration signatures should be more distinct for the different regions. For example, N1 and N2 should both be affected by X1 while N3-N5 would be largely unaffected. Also, N3-N5 should all be affected by X3. Some potential confusions would be X2 and X3 as well as X0 and X2.

Figure 34. Diagram of Room C showing regions (X#) for context-based localization

Table 32. Summary statistics from context calibration of Room C

		Node														
		N1			N2			N3			N4			N5		
		n_e	$\overline{R}_e$	σ_e	n_e	$\overline{R}_e$	σ_e	n_e	$\overline{R}_e$	σ_e	n_e	$\overline{R}_e$	σ_e	n_e	$\overline{R}_e$	σ_e
Region	X1	349	59.0	2.0	349	57.1	5.9	360	58.6	5.4	362	52.9	7.7	356	61.2	1.7
	X2	335	59.7	1.3	382	58.2	6.0	392	58.9	5.3	349	47.2	6.0	360	58.6	8.8
	X3	313	59.5	2.0	346	56.7	7.1	373	58.4	5.7	376	47.8	7.2	385	56.4	11.3

Calibration.

Calibration for Room C was exactly the same as that of Room A. Fifteen minutes were allocated and divided evenly for all three regions. In each trial, the target was positioned and oriented randomly in the region. Collections were each five minutes long with pauses every minute to change position and orientation. The resulting offline data map signatures are shown in Table 32.

Interestingly, there was a lot more variation in this room among the three regions and a lot of the expected behaviors were observed. N1 and N3 both observed similar $\overline{R}_e$ across all three regions. N1 noted a minimum for X1, which was expected since N1 was

in X1. N3 observed a lot of variation in all three cases, which is most likely due to the absorption from the bed and reflections elsewhere.

N2, N4, and N5 all recorded the primary behavior of high $\overline{R}_e$ during X1. N4 and N5 also recorded lower values for both X2 and X3. N4 observed large variation in all cases, most likely because it is in an area that would receive a lot of reflections. However, the important observation was that for X2, this variation was smaller as the target focused on attenuation signals in the area that would reach N4. For X3, variation in N4 increased again, but $\overline{R}_e$ was still smaller than that of X1. Similarly, N2 and N5 both observed smallest values with high variations for X3. This high variation is most likely due to their distance from the transmitting Hub and how, despite the small size of the region, the target's orientation caused great fluctuations in the recorded values from these nodes.

Therefore, these calibration results were more descriptive than expected from previous scenarios. However, the high fluctuations detected by the system may still produce performance results that are no better than previously seen. The absorbent materials seemed to also be a cause for high fluctuations in the system as some signals were slightly attenuated and paired with reflected signals.

Observation Results.

Twenty trials were conducted in Room C to observe the localization performance in this room. Five trials of one minute each were allocated for each of the four possible regions. The interval for events was still $\tau = 20$ s. Therefore, there was a total of 60 events again. The results from each trial are shown in Table 44 in Appendix B. In this section, the confusion matrices in Table 33 summarize these results.

X1 was never mistakenly selected as the location of a target in X2, which supports how different the signatures at X1 and X2 were. However, X2 was mistakenly selected as the location of a target in X1 several times, most likely because in both cases only one

Table 33. Confusion matrices with varying g for context-based localization in Room C

		Actual Region				
		X1	X2	X3	X0	
Predicted	X1	5	0	4	1	10
	X2	4	8	7	6	25
	X3	6	7	4	8	25
	X0	0	0	0	0	0

(a) $g = 0$

		Actual Region				
		X1	X2	X3	X0	
Predicted	X1	5	0	4	1	10
	X2	4	8	7	6	25
	X3	6	7	4	8	25
	X0	0	0	0	0	0

(b) $g = 1$

		Actual Region				
		X1	X2	X3	X0	
Predicted	X1	5	0	3	1	9
	X2	4	8	8	6	26
	X3	6	7	4	8	25
	X0	0	0	0	0	0

(c) $g = 2$

		Actual Region				
		X1	X2	X3	X0	
Predicted	X1	5	0	3	1	9
	X2	4	7	8	6	25
	X3	6	7	4	8	25
	X0	0	1	0	0	1

(d) $g = 3$

		Actual Region				
		X1	X2	X3	X0	
Predicted	X1	4	0	2	2	8
	X2	5	7	5	5	22
	X3	5	7	5	7	24
	X0	1	1	3	1	6

(e) $g = 4$

		Actual Region				
		X1	X2	X3	X0	
Predicted	X1	0	0	0	0	0
	X2	3	5	4	4	16
	X3	4	1	2	4	11
	X0	8	9	9	7	33

(f) $g = 5$

node was greatly affected so RSS fluctuations confused these regions.

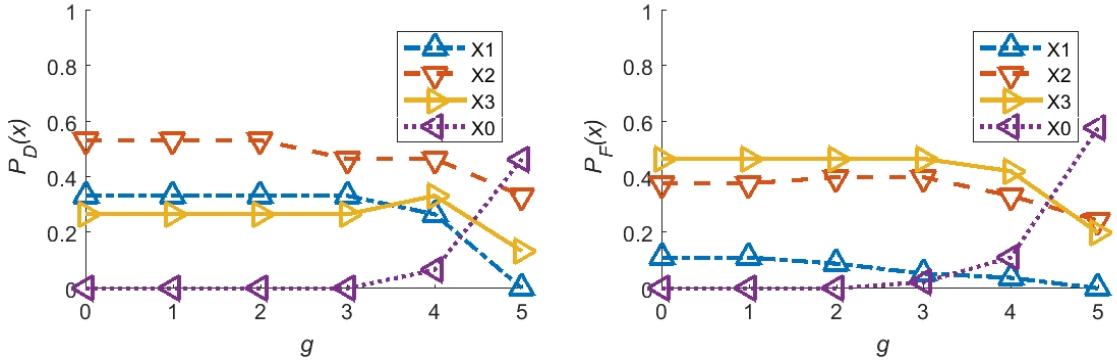
It appeared, however, that $x_p = 2$ and $x_p = 3$ were both very frequent and that these regions were often confused with each other. As g increased, the frequency of $x_p = 1$ was relatively the same and often successfully localized a target in that region. This supports how unique X1's signature was to the other two regions. However, when $g = 5$, this region was never selected, which suggests that fluctuations in the system made it difficult to select X1 when all nodes had to achieve $P(1,n) > \alpha$.

The empty case selection was again non-existent with lower g but was dominant at 33 times when $g = 5$. This was 55.0% of the total selections though, and X2 and X3 were still both frequently selected.

Table 34 displays the individual region performance metrics from these results. The trends depicted in Figures 35 and 36 were expected with increasing g, but $P_D(x)$ remained high with similar $P_F(x)$. The most probable selections of $x \neq 0$ performed fairly well and evenly so. The ratio of $P_D(1)$ to $P_F(1)$ was impressive when $g \leq 4$. On the other hand, X2 and X3 both saw high $P_F(x)$. While X2 was also able to achieve high $P_D(2)$ up

Table 34. Performance by region $(P_D(x), P_F(x))$ for context-based localization in Room C using threshold requirements (g)

	$g=0$		$g=1$		$g=2$		$g=3$		$g=4$		$g=5$	
Region	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$	$P_D(x)$	$P_F(x)$
X1	33.3%	11.1%	33.3%	11.1%	33.3%	8.9%	33.3%	5.3%	26.7%	3.8%	0.0%	0.0%
X2	53.3%	37.8%	53.3%	37.8%	53.3%	40.0%	46.7%	40.0%	46.7%	33.3%	33.3%	24.4%
X3	26.7%	46.7%	26.7%	46.7%	26.7%	46.7%	26.7%	46.7%	33.3%	42.2%	13.3%	20.0%
X0	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.2%	6.7%	11.1%	46.7%	57.8%
Average	28.3%	23.9%	28.3%	23.9%	28.3%	23.9%	26.7%	23.6%	28.3%	22.6%	23.3%	25.6%



(a) $P_D(x)$ by Region (b) $P_F(x)$ by Region

Figure 35. Effect of varying g on region performance $(P_D(x), P_F(x))$ in Room C

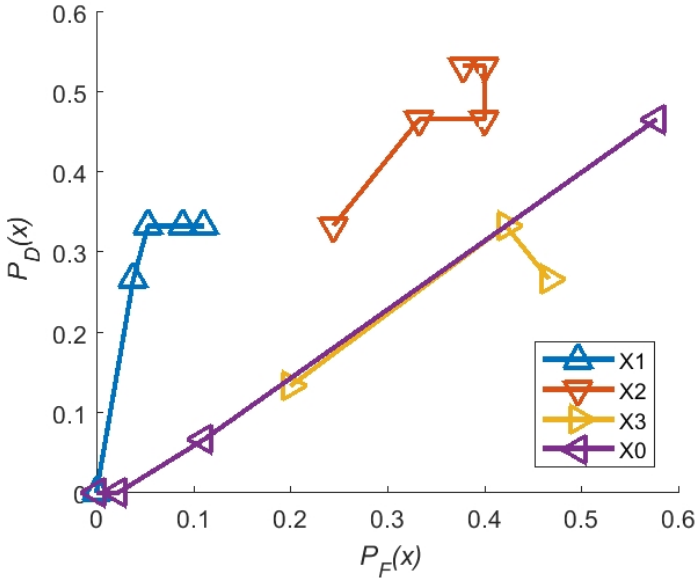


Figure 36. ROC graph for context-based region performance $(P_D(x), P_F(x))$ with varying g in Room C (for X1-3, left to right is decreasing g; for X0, left to right is increasing g)

to 53.3%, X3 achieved $P_D(3) \leq 33.3\%$. Therefore, these results indicate that confusion of other regions for $x_{true} = 3$ was frequent despite $x_p = 3$ being so prevalent.

On average, performance was relatively the same in all cases, but the best performance was again when $g = 4$ when the average true and false positives were average $P_D(x) = 28.3\%$ and average $P_F(x) = 22.6\%$. While average $P_D(x)$ for $g \leq 4$ were better than the 25.0% of random guessing, they were not much greater. When $g = 5$ though, the average $P_D(x) = 23.3\%$ was less than 23.3% as more false positives were reported and no X1 targets were accurately localized. This suggests that the absorbent materials in the room made localization using this system difficult due to high fluctuations.

Interestingly, $P_F(x)$ remained low compared to results seen in the other scenarios; it never went above 46.7% (for X3). The increase of $x_p = 0$ frequency due to increasing g was also high, but it completely reduced $P_D(1)$ to 0.0%. This is most likely because X1 depended largely on just one node and so requiring $g = 5$ when so much variation could occur in the room made it difficult for $x_p = 1$. In fact, according to the confusion matrix, all other regions were preferred over X1.

Table 35 displays the accuracy and overall performance metrics. P_D and P_A were low when $g = N$ and while P_F did decrease from 100.0%, it only decreased to 53.3%. This decrease though was largely due to the surge in $x_p = 0$ that eliminated $x_p = 1$. Low P_A was due to a large part because of the fluctuations seen in the room. These results suggest that a room with absorbent materials may benefit from more receiver nodes or a method that weighted different nodes for different regions.

5.4.3 Comparison of Results.

All three scenarios of deploying the context-based method were met with greater P_A and lower P_F than the cell-based method. Additionally, calibration time was much lower, which allowed for more time to calibrate each region for a better offline data map.

Table 35. Accuracy (P_A) and overall localization performance (P_D, P_F) for context-based localization in Room C using threshold requirements (g)

g	P_D	P_F	P_A
0	37.8%	100.0%	28.3%
1	37.8%	100.0%	28.3%
2	37.8%	100.0%	28.3%
3	35.6%	100.0%	26.7%
4	35.6%	93.3%	28.3%
5	15.6%	53.3%	23.3%

In all three scenarios, P_A achieved was greater than random guessing (except when $g = 5$ for Room C). However, results were still not ideal.

Figure 37 depicts the P_A trend in each room with respect to changing the threshold parameter g. When $g = N$, low $P_F(x)$ was found in all cases as $x_p = 0$ became more frequent. While Room B was higher for all g, this was largely due to the fact that there were only three possible regions to select from as opposed to the four in Rooms A and C. In all three rooms, P_A remained the same when $g \leq 2$. In fact, individual localization results $(P_D(x), P_F(x))$ did not change in this interval of g. This suggests that g more greatly affects results when g was closer to N.

In all scenarios, P_A decreased as g increased from 4 to 5. This was largely due to $x_p = 0$ being more frequent and raising the true negatives but often at the expense of true positives for non-X0 regions. Interesting behavior occurred in Rooms B and C, where P_A at $g = 4$ was highest across all g. This occurred during a sort of compromise area where all possible selections were observed instead of that observed when $g = 5$. Room A did not observe this compromise area, possibly because the setup of the regions in this scenario better affected calibration of the nodes.

Despite the fact that Rooms A and C both had three regions, Room A outperformed Room C in P_A. This suggests that the fluctuations of RSS due to absorbent materials confused regions in Room C with metal furniture as opposed to the empty Room A.

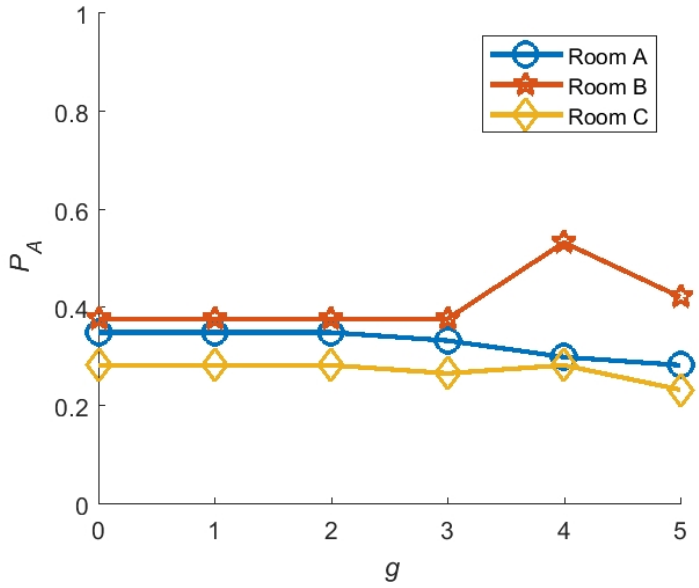


Figure 37. Comparison of accuracy performance (P_A) in Rooms A, B, & C with varying g

If Room B were calibrated for three scenarios instead of two and then tested similarly to Rooms A and C, its P_A might be slightly worse than that of either other scenario at low g, but with higher g, the lack of absorbent materials and the presence of metal objects that help reflect signals might produce better results.

Other confounding variables include the position of the nodes with respect to the context regions. For example, nodes in corners may receive more fluctuation than nodes along wall edges or in the open. Nodes not in regions or not in a position for its LOS to be blocked by a target in the region would not be able to produce differing results from other regions and so would be unhelpful in localization. The exception would be when the target was not in any region and so the empty case would have been reported should this node have been activated when it was never activated for other regions.

Figure 38 is a ROC graph that depicts the overall performance metrics for P_D and P_F in the three different scenarios. Room B was able to record both ends of the P_F range, and P_D decreased the least in this scenario over the range of g. This only confirms that the system using many fewer regions would of course perform better. The scenarios with

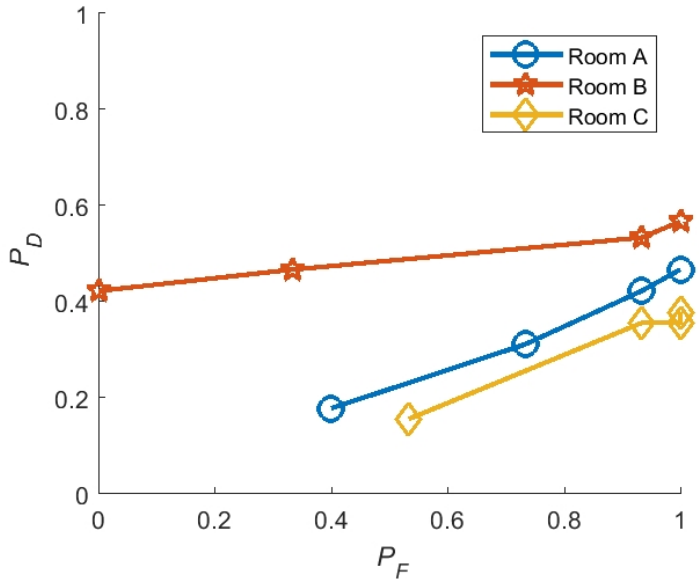


Figure 38. ROC graph comparing the overall performance (P_D, P_F) in Rooms A, B, & C with varying g (left to right, g decreases)

three regions were more interesting, but the results were often worse. Room A was only able to achieve P_F as low as 40.0% but also observed a lowered P_D from 46.7% to 17.8%, which was worse than random guessing. For Room C, in order to lower P_F to 53.3%, P_D went from 37.8% to just 15.6%.

A large factor was that the method did not allow the empty case to be reported unless certain parameters were met. What was not expected was that most, if not all, non-X0 regions actually were able to produce valid $P(x,n)$ for selection. Therefore, X0 was not more frequently selected, especially with lower g. This would be a future investigation of the system to reduce P_F while also not sacrificing P_D as heavily.

In conclusion, deploying this method across all three scenarios resulted in relatively similar results. This supports the idea that fingerprinting can be deployed in many situations indiscriminately. However, a smaller number of context regions would improve P_A. Also, a room with a lot of absorbent materials may also fail to achieve lower P_F as a result of large fluctuations.

5.5 Summary

This chapter deployed and evaluated the RSS DfP system for both detection and localization. Detection using a simple setup was highly favorable. Signal attenuation at the midpoint of the empty test room was 4.0 ± 0.8 dB. At six other points along the line of sight, signal attenuation at different distances ranged from 2.2 to 4.4 dB. Theoretically, the target should have attenuated signals more as it approached either node, but the inverse was observed; $\overline{R}_a(d)$ was unexpectedly smaller near either node. This could be explained by a high f_s that was calibrated using the average maximum R_s. Despite this unexpected behavior, detection using the chosen method could work.

Detection experiments were conducted at seven points along the LOS. P_D was high in all cases; at every distance except d = 1.0 ft, $P_D \geq 99.6\%$. At d = 1.0 ft, P_D was at its lowest, but was still 81.8%. However, as $\overline{R}_a$ was smaller at distances closer to either node, P_F was also higher. P_F reached up to 53.2% at d = 1.0 ft and was 51.0% and 49.6%, respectively at d = 7.5 and 9.0 ft. This was a huge difference from the low values observed near the center of the room ($1.6\% \leq P_F \leq 6.4\%$). Again, this could be due to the high f_s that did not allow the system to collect smaller RSS.

Localization was tested to expand the scope of the detection problem. However, cell-based localization performed poorly. In fact, $P_A \leq 5.4\%$ and $P_F \geq 88.3\%$, both of which came from two different sets of orientation data. As a result, two ideas were investigated to improve upon this method.

To lower P_F, g was introduced to force a candidate location to have a certain number of nodes that returned a p-value $> \alpha = 0.05$. In the cell-based experiment, this was able to lower $P_F = 96.3\%$ to just 42.3%. To raise the low P_A, the criterion for a correct selection was expanded to adjacent cells. As a result, the $\overline{P_A}$ of all the data sets used went from just 4.0% to 22.2%, so limiting the number of possible locations could improve P_A. The context-based method adapted the cell-based method with these two ideas.

The context-based method was first deployed in the same room as the previous experiments and only calibrated three locations of interest. As a result,when $g = 5$, the accuracy was $P_A = 28.3\%$ with only $P_F = 40.0\%$. Therefore, this method improved upon the cell-based method. Additionally, whereas the cell-based method was calibrated in almost two hours, the new method only used fifteen minutes.

To test the robustness of the system for localization, experiments were conducted in two more typical rooms: a computer room and a bedroom. The results were fairly comparable in all cases. At $g = 5$, Room B achieved $P_A = 42.2\%$ and $P_F = 0.0\%$, and Room C achieved $P_A = 23.3\%$ and $P_F = 53.3\%$. The greater success of Room B however, could be attributed to having only two regions instead of three like Rooms A and C had. Also, Room C's poorer performance could be attributed to the absorbent materials in that room causing large fluctuations that made it difficult to achieve discriminating calibration data for the different regions.

This chapter showed high success of the detection system. The cell-based localization method was found to have serious flaws, and so a new method was developed to improve upon it. As expected, this context-based method improved the localization performance. Despite not achieving optimal localization performance, the results showed promise for future DfP system deployment on top of home automation RF devices.

VI. Summary and Conclusions

The research presented in this thesis aimed to develop a proof of concept for a DfP system capable of detection and localization that exploits the RF signals transmitted in a home automation network.

Chapter II presented background information and a literature review of this topic. Various techniques for human detection and localization were discussed. Ultimately, a radio-based, device-free, and passive technique was determined to be ideal for indoor localization because of its low cost, covert performance, and ability to be installed on top of home automation networks. Such a technique can exploit signals used in a home automation network and detect changes in the environment to passively detect the target without requiring the target to carry any devices.

Several methods for detection and localization were also presented. RSS was preferred over ToA or AoA for DfP techniques because the measurement was easy to measure and the signals already existed in wireless home automation networks, which lowered costs. For one-receiver, DfP detection, the baseline signal comparison approach was chosen for its simplicity. For multiple-receiver, RSS-based DfP localization, the probabilistic fingerprinting method was chosen since it outperformed deterministic methods and could, unlike RTI, be used in networks with fewer nodes. The centralized approach was also chosen to allow all receiving nodes to share information and thus increase accuracy.

Related works were reviewed to further inform the selection of different techniques and methods. The focus of current indoor localization research was observed to mostly improve upon existing methods and to choose a best method. However, none of the reviewed works deployed their system on devices specifically intended for home automation. Doing so, however, could generalize the DfP method for farther reach.

Insteon devices were chosen among the home automation technologies because of their simulcasting feature, because they operate on the less-crowded 915 MHz band, and because of the availability of papers detailing its technology. However, an investigation of the technology revealed that Insteon devices did not record RSS. Therefore, a cheap alternative was sought for the proof of concept.

Chapters III and IV presented various topics that contributed to the development of the DfP detection and localization system. SDR receivers were used as the alternative for recording RSS. The Insteon network was constructed using a Hub and Keypads. Since dual-band Insteon devices sent out RF messages only on the powerline zero crossing, scripts were developed to generate Insteon messages for collection by exploiting the Hub's Internet connectivity and to incorporate into the algorithm a filter threshold of the silent periods when the next messages were waiting for the next zero crossing.

Chapter V then deployed this system to test its capabilities and limitations for detection and localization. Detection was performed using a simple setup tripwire consisting of one Insteon Hub and one receiver node. Localization was tested first using a cell-based method. After the poor results, a context-based method was developed and then tested in the same room as well as two other rooms.

6.1 Conclusions of Research

The proof-of-concept system was developed as an RSS-based DfP system capable of detection and localization. The system in this research used Insteon home automation devices and attempted to exploit the RF signals transmitted between these devices on the 915 MHz frequency band.

When deployed, the system performed very well in detection. $\overline{R}_a(d)$ was observed to range from 2.2 to 4.4 dB depending on the LOS distance from the Insteon Hub. Detection experiments were conducted at seven LOS distances. A detection event occurred if

an observed RSS value in a one-second interval was between a threshold set during calibration. In all cases, $P_D \geq 81.8\%$, and even $P_D = 100.0\%$ was observed. Some distances suffered from high P_F (up to 53.2%). This was largely because of the smaller thresholds as a result of smaller $\overline{R}_a$. However, when $\overline{R}_a$ was higher, P_F was as low as 1.6%.

The cell-based localization method did not perform as well. In fact, P_A was only as high as 5.4%, and the lowest $P_F = 88.3\%$. When the method was adapted to include a threshold parameter requiring a certain number of nodes for a selected cell return a p-value greater than 0.05, P_F was able to lower to just 42.3%. To improve P_A, a proximity argument for true positives was added and was able to raise the average P_A from just 4.0% to 22.2%.

As a result of these improvements, the context-based method was developed with a threshold parameter and a limit to the number of possible locations. These locations were a few context regions instead of many single cells that spanned the entire room. When deployed with this new method to look for just three context regions, the system was able to achieve $P_A = 28.3\%$ and just $P_F = 40.0\%$. In addition, calibration time went from two hours to just fifteen minutes. Therefore, significant improvements were developed on the localization system for the proof of concept.

To test the robustness of the system in other rooms that better represented potential areas of deployment, the system was deployed in a computer room and a bedroom. The results were comparable in all three rooms. Room B, with just two context regions, achieved $P_A = 42.2\%$ and perfect $P_F = 0.0\%$. Room C achieved $P_A = 23.3\%$ and $P_F = 53.3\%$, the worst of the three results, but this suggested that absorbent materials like beds and pillows could cause large fluctuations that made it difficult to achieve distinct calibration data for the different regions. Despite not achieving optimal performance in any case, the results showed promise for future DfP system deployment on top of home automation RF devices.

6.2 Significance of Research

This research sought to provide a proof of concept for a device-free passive system capable of detection and localization of a target through the exploitation of a home automation network's radio frequency signals. The proof of concept was developed using Insteon devices and the associated IoT home automation network. Although using Insteon technology had a few limitations for DfP detection and localization, investigations resulted in creating tools that allowed for a proof-of-concept system that limited costs and performed covertly. Ultimately, this research demonstrated the feasibility of exploiting a home automation network for DfP indoor detection and localization.

During deployment and testing, the system performed very well in detection, but localization provided less-than-desirable results, even with an improvement in the methods. Nevertheless, the research was able to demonstrate that a cell-based fingerprinting method performs worse than a context-based fingerprinting method. Thresholds for fingerprinting were also demonstrated to reduce P_F. Furthermore, similar results in three different scenarios demonstrated the robustness of the system, while differences demonstrated potential effects of different rooms.

6.3 Recommendations for Future Research

Despite these achievements and advancements, this research would like to suggest future avenues of research and development for a better proof-of-concept system.

Development with Insteon.

Several limitations were encountered during the development of the proof of concept using Insteon devices. Chief among these were no intrinsic ability to compute RSS using just Insteon devices and the inconvenience of waiting for the zero crossing between transmissions. These limitations required the proof of concept in this research

to use separate devices to collect RSS and to incorporate an unreliable silent period filter threshold into the collection algorithms. An ideal proof of concept would use only Insteon devices for collection of RSS and could do so without having to mind silent periods.

The system should be deployable on top of existing devices and thus eliminate the current DCPS unit, reducing monetary and installation costs further. The Insteon devices could individually collect RSS from other devices and send them to a central Insteon node like the Hub. When prompted from the user, this central node could run the DfP program and report to the user the results via the mobile app or even automatically control appliances and other pre-programmed settings.

One avenue towards achieving such a system would be to develop this system as an Insteon developer. Becoming an Insteon developer requires applying to the company, a process which could take weeks. However, doing so could allow modifications to the Insteon devices and API, and additions to the Insteon protocol to overcome these limitations for DfP purposes. For example, a command may be added that sends discrete messages for DfP purposes only from dual-band devices without needing to synchronize with the powerline zero crossing. These messages could contain RSS from the nodes. Localization performance could improve greatly as a result.

Number of Devices.

While Section 2.3 discussed related works that used up to sixteen nodes to achieve their high performance results, this research attempted to perform accurate localization using only three or five nodes. The reasoning behind this choice was to simulate a typical home automation network, but with increased home automation prevalence, the number of nodes in a room could be higher. Therefore, future research may increase the number of nodes to achieve better results.

Silent Period Filter Threshold.

A silent period filter threshold was calibrated to minimize the RSS captured when the Insteon signal was not transmitting. But, in calculating the signal attenuation levels for detection, a behavior inverse to what was expected was observed. Although the effect of the filter threshold should not have mattered using the detection method, a possible explanation was the high f_s calibrated. Thus, investigating a method to eliminate the silent period could greatly improve the system.

Weighting Algorithm.

Another future recommendation for research based on experiment results is a weighting algorithm to improve P_A. Currently, the method described in the proof of concept only enforced a requirement on a number of nodes that must produce $P(c,n) > 0.05$. However, this does not discriminate which nodes for a particular region and can accept a maximum probability from any group of nodes. For example, a target in X2 of Room C may be better localized if just $P(2,4)$ and $P(1,4)$ were high as opposed to the current method that would accept $P(3,4)$ and $P(5,4)$ for meeting the requirement. In general, the RSS DfP localization method could be improved.

Real-Time Analysis & Other Script Improvements.

A limitation of this system was that collection and location selection occurred separately. Ideally, these two processes would be performed in real time such that a location would be reported as collection was occurring. This was acceptable for purposes of the proof of concept, but with improved methods and better incorporation of the system into devices, this would be a definite consideration. This location selection should be done at the central hub or on some cloud server since individual Insteon devices may not be equipped to handle such computations.

Scripts were also largely experiment-specific. For applications, live collection should loop for as long as the user wishes to perform detection or localization and would continuously print out results. Also, since the DCPS would already be connected to the Internet for message generation, the script could alternatively print the output in other places, including social media like in ref. [23]. In detection, the script should also be able to detect along the LOS without inputting d.

Power Consumption.

Another consideration for future development is power consumption, especially due to message generation. If the system were deployed on only plug-in or wall devices, this would not be as big of an issue. But for devices that are remote or are for example LED bulbs, having additional functions for DfP detection and localization may decrease the life expectancy of these devices or cause large increases in electricity billing. Message generation should only be used when detection and localization is prompted, but to have message generation also send messages that perform another important function like checking the statuses of devices, perhaps this would not be an issue.

Other Home Automation Protocols.

As discussed in the background section, a popular frequency band for indoor localization is the 2.4 GHz band. A DfP system that exploits this band may not have to worry about silent periods or message generation since the band is often highly congested from various devices. Thus, exploiting these RF signals for RSS DfP detection and localization would not necessarily require message generation scripts. In addition, some of the other home automation protocols may have a field for RSS, or their devices may be programmable to include these recordings. A consideration, however, is that there is a lot of interference in this band, which could cause fluctuations that affect the RSS.

Appendix A. Python Script for On-Demand Insteon RF Message Field

The following Python 2.7 code generates a field of 915 MHz signals containing Insteon command messages to an unlinked device. It requires a connection to the Internet and for the network to be the same as that of the Insteon Hub. Authentication credentials are required the first time this script is initiated.

```
import sys
import argparse
import urllib2
from bs4 import BeautifulSoup
import base64
import socket
import time
import pathlib2 as pathlib

def init():
    global hub_address
    global repeat_send
    global auth_creds
    global delay

    parser = argparse.ArgumentParser(add_help=True,
                                     epilog='example:\n\t#./InsteonMsgGeneration.py -s 1234AB -c 100',
                                     description='send Insteon command to Hub over HTTP')

    parser.add_argument('-s', '--src', dest='src',
                        default=None,
                        help='Insteon hub address (IP:port); if empty, program will check network for hub')

    parser.add_argument('-a', '--auth', dest='auth',
                        help='hub authentication credentials of form user:pass')

    parser.add_argument('-c', '--count', dest='count',
                        type=int,
                        help='number of times to send packet (can cancel with Ctrl+C)')

    parser.add_argument('-t', '--time', dest='time',
                        type=float, default=0.5,
                        help='delay in seconds between sending messages')

    args, unknown_args = parser.parse_known_args()

    if args.src:
        hub_address = args.src

    if args.auth:
        auth_creds = args.auth

    if args.count:
        repeat_send = args.count

    if args.time:
        delay = args.time


def grab_hub_info():
    resp = urllib2.urlopen('https://connect.insteon.com/getinfo.asp')
    page = resp.read()
    soup = BeautifulSoup(page,'lxml')
```

```
54      tables = soup.findAll('table', attrs={'bgcolor': '#FFFFFF'})
55      count_hubs = len(tables)
56      if count_hubs == 0:
57          print 'No Hubs have been connected to this network'
58          sys.exit()
59      rows = []
60      for table in tables:
61          rows.append(table.findAll('tr'))
62      all_get_info = []
63      for i in range(count_hubs):
64          get_info_header = []  #
65          get_info_value = []  # all values for single Hub
66          for row in rows[i]:
67              j = 1  # counter
68              for td in row.find_all('td'):
69                  # get text
70                  # table has three columns; c1 is heading, c2 is space, c3 is value
71                  if j % 3 == 1:
72                      get_info_header.append(td.get_text().encode('ascii').strip('\n'))
73                  elif j % 3 == 0:
74                      get_info_value.append(td.get_text().encode('ascii').strip('\n'))
75                  j += 1
76          get_info_dict = zip(get_info_header, get_info_value)  # tuple of single Hub's additional info
77          all_get_info.append(get_info_dict)  # append to all info list
78      all_internal_ips = []
79      for link in soup.find_all('a'):
80          all_internal_ips.append(link.get('href'))
81      all_hubs_info = zip(all_internal_ips, all_get_info)  # tuple listing each hub's IP and info
82      good_hubs = []
83      for i in range(len(all_hubs_info)):
84          is_configured_connected = all_hubs_info[i][1][0][1]
85          if is_configured_connected == 'TRUE':
86              good_hubs.append(all_hubs_info[i])
87      if len(good_hubs) == 0:
88          print 'No Hubs connected to an account'
89          sys.exit()
90      good_hubs_info = good_hubs
91      all_ips_and_ids = []
92      for i in range(len(good_hubs_info)):
93          ip_and_id = [good_hubs_info[i][0], good_hubs_info[i][1][12][1]]
94          all_ips_and_ids.append(ip_and_id)
95      return all_ips_and_ids
96
97
98  def send_message(hub_ip, hub_port, hub_id, msg, nmsg):
99      s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
100     s.connect((hub_ip, hub_port))
101     s.send(msg)
102     while 1:
103         response = s.recv(1024)
104         if '200' in response:  # OK
105             break
106         else:  # Failed
107             sys.exit(response)
108     s.close()
109
110     return 'Hub ' + hub_id + ' sent ' + str(nmsg) + ' message(s)\n'
111
112
113 def scrape_getinfo():
114     info_good_hubs = grab_hub_info()  # scrape Insteon site for Hubs connected and configured on local network
115     [hub_url, hub_id] = [info_good_hubs[0][0], info_good_hubs[0][1]]  # just the first hub
116     good_hub_info = [hub_url, hub_id]
117     return good_hub_info
118
119
```

```
120 def save_info(destination_set, auth_set, hub_ip, hub_port, hub_id, credentials):
121     if destination_set == False:
122         new_hubs_file = open('prevHubConnect', 'w')
123         new_hubs_file.write(hub_ip + ':' + str(hub_port) + ' ' + hub_id)
124         new_hubs_file.close()
125     if auth_set == False:
126         new_hubs_file = open('prevCreds', 'w')
127         new_hubs_file.write(credentials)
128         new_hubs_file.close()
129 
130 
131 def main():
132     init()  # initialize global variables
133 
134     # set source
135     destination_set = False
136     try:
137         hub_address
138     except NameError: # no user-inputted hub address, check file
139         previous_hubs_file = pathlib.Path('prevHubConnect')
140         if previous_hubs_file.is_file():
141             previous_hubs = open('prevHubConnect','r').readline() # parse file for first previous hub
142             if previous_hubs == '': # empty file
143                 [hub_url, hub_id] = scrape_getinfo()
144             else: # split the line for the url and id
145                 destination_set = True
146                 [hub_url, hub_id] = previous_hubs.split(' ',1)
147         else: # not valid file, scrape web page
148             [hub_url, hub_id] = scrape_getinfo()
149     else:
150         hub_url = hub_address
151         hub_id = 'Your Hub'
152     [hub_ip, hub_port] = hub_url.strip('http://').split(':')  # split Hub URL into IP and port
153 
154     # encode credentials
155     auth_set = False
156     try:
157         auth_creds
158     except NameError:
159         previous_hubs_file = pathlib.Path('prevCreds')
160         if previous_hubs_file.is_file():
161             previous_creds = open('prevCreds', 'r').readline()  # parse file for first previous hub
162             if previous_creds == '':  # empty file
163                 sys.exit('No authorization credentials found')
164             else:  # split the line for the url and id
165                 auth_set = True
166                 these_auth_creds = previous_creds
167         else:  # not valid file, scrape web page
168             sys.exit('No file for authorization credentials found')
169     else:
170         these_auth_creds = auth_creds
171     auth_encode = base64.b64encode(these_auth_creds)  # encode basic authentication credentials
172     auth = 'Authorization: Basic ' + auth_encode + '\r\n '
173 
174     # craft GET message
175     msg = 'GET /3?02620000000F1000=I=3 HTTP/1.1\r\n ' \
176           + 'Cache-Control: max-age=0\r\n' \
177           + auth + '\r\n\r\n'
178 
179     # send message(s) and print result to screen
180     try:
181         repeat_send
182     except NameError:
183         try:
184             n = 1
185             while True:
```

```
186                     print send_message(hub_ip, int(hub_port), str(hub_id), msg, n)
187                     time.sleep(delay)
188                     n += 1
189             except KeyboardInterrupt:
190                 pass
191     else:
192         try:
193             for nrepeat in range(repeat_send):
194                 print send_message(hub_ip, int(hub_port), str(hub_id), msg, nrepeat+1)
195                 time.sleep(delay)
196         except KeyboardInterrupt:
197             pass
198     save_info(destination_set,auth_set,hub_ip,hub_port,hub_id,these_auth_creds)
199
200 if __name__ == '__main__':
201     main()
202     exit(0)
```

Appendix B. Additional Data from Experiments

Table 36. Results for event period calibration for LOS detection

Trial	n_e	$\overline{R}_e$ (dB)	σ_s (dB)
1	372	58.3	1.0
2	374	58.3	1.1
3	380	58.4	1.4
4	384	58.4	1.1
5	385	58.4	1.2
6	378	58.5	0.8
7	383	58.6	0.9
8	376	58.6	1.2
9	375	58.7	0.8
10	376	58.8	0.9
11	381	58.9	1.0
12	382	58.9	1.1
13	389	59.1	0.8
14	371	59.1	1.4
15	375	59.5	1.0
Avg		58.7	1.1

(a) $d = 1.0$ ft

Trial	n_e	$\overline{R}_e$ (dB)	σ_s (dB)
1	354	56.6	0.8
2	363	56.5	0.9
3	360	56.4	0.9
4	364	56.6	1.0
5	372	56.6	0.8
6	362	56.5	1.2
7	369	56.6	0.8
8	365	56.6	0.9
9	371	56.5	1.0
10	378	56.6	1.2
11	389	56.5	1.5
12	373	56.7	0.9
13	387	56.7	0.9
14	361	56.8	1.0
15	370	56.7	1.2
Avg		56.6	1.0

(b) $d = 2.5$ ft

Trial	n_e	$\overline{R}_e$ (dB)	σ_s (dB)
1	361	57.1	1.3
2	353	56.9	1.2
3	375	57.1	1.1
4	360	57.0	1.2
5	372	56.9	1.4
6	363	57.0	1.1
7	373	57.0	1.0
8	366	57.0	0.8
9	362	57.0	0.9
10	367	56.9	1.1
11	380	57.0	1.1
12	366	57.0	0.9
13	380	57.0	1.3
14	377	57.2	0.9
15	376	57.0	0.9
Avg		57.0	1.1

(c) $d = 4.0$ ft

Trial	n_e	$\overline{R}_e$ (dB)	σ_s (dB)
1	298	57.2	1.1
2	344	57.1	1.0
3	372	57.3	1.0
4	368	57.3	1.2
5	368	57.3	1.0
6	373	57.3	1.5
7	364	57.3	1.1
8	375	57.4	0.9
9	365	57.3	0.9
10	373	57.4	0.9
11	376	57.3	0.9
12	367	57.3	1.3
13	374	57.4	1.1
14	363	57.3	1.2
15	379	57.2	1.1
Avg		57.3	1.1

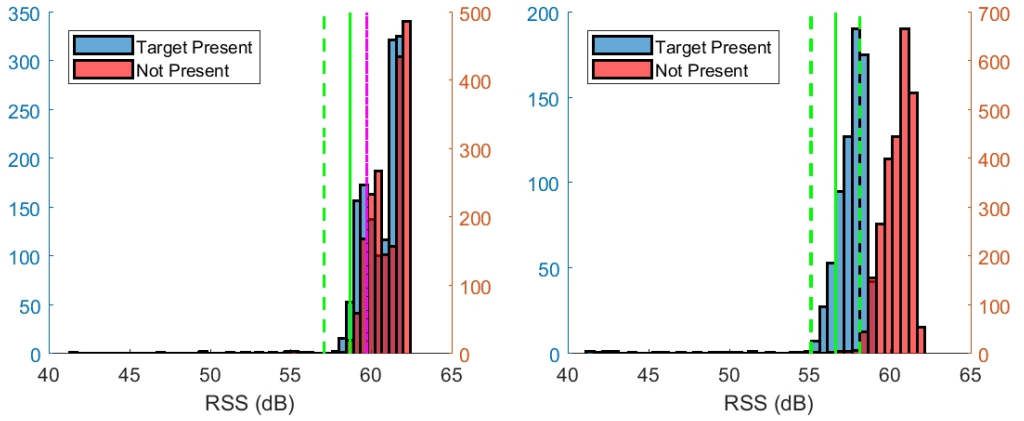
(d) $d = 6.0$ ft

Trial	n_e	$\overline{R}_e$ (dB)	σ_s (dB)
1	367	58.7	0.8
2	365	58.7	0.9
3	377	58.7	1.1
4	360	58.7	0.8
5	359	58.7	0.8
6	374	58.6	1.2
7	370	58.7	0.9
8	374	58.7	1.2
9	372	58.5	1.3
10	355	58.4	1.0
11	363	58.5	0.9
12	377	58.5	1.2
13	377	58.5	1.4
14	378	58.6	1.1
15	383	58.6	1.0
Avg		58.6	1.1

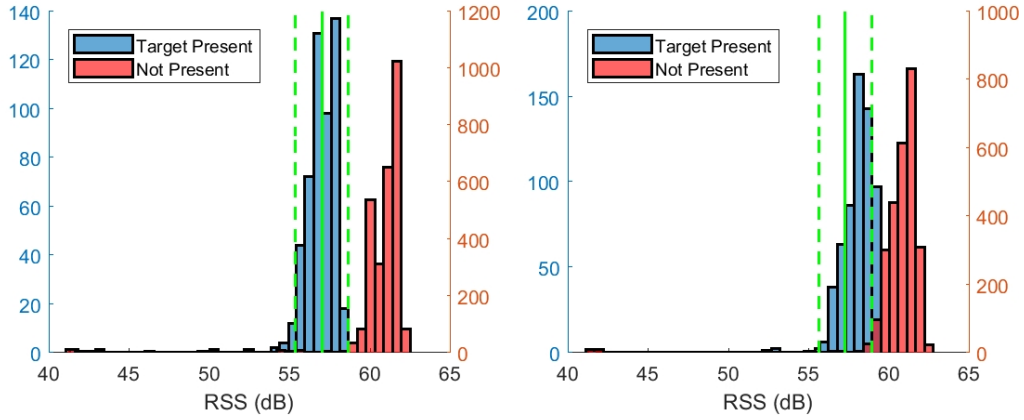
(e) $d = 7.5$ ft

Trial	n_e	$\overline{R}_e$ (dB)	σ_s (dB)
1	377	59.0	1.1
2	378	59.1	1.1
3	380	59.1	1.0
4	374	59.1	1.1
5	372	59.1	1.1
6	352	58.8	0.9
7	361	58.7	1.4
8	358	58.7	1.0
9	369	58.8	1.1
10	366	58.8	1.2
11	370	58.8	1.3
12	356	58.9	0.9
13	364	58.9	1.2
14	354	58.8	1.5
15	358	59.0	1.4
Avg		59.1	1.2

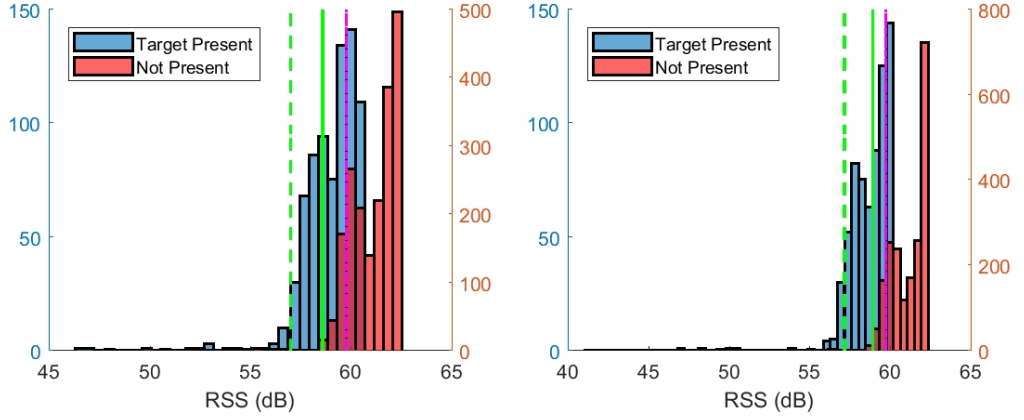
(f) $d = 9.0$ ft



(a) $d = 1.0$ ft (b) $d = 2.5$ ft



(c) $d = 4.0$ ft (d) $d = 6.0$ ft



(e) $d = 7.5$ ft (f) $d = 9.0$ ft

Figure 39. Histograms of RSS distribution from detection collection at different distances, d

Predicted Cell of Target	Actual Cell of Target																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
1	0	3	2	3	2	0	1	3	3	1	11	1	2	5	2	2	2	3	1	1	0	0	0	0	0	0	48
2	1	2	2	0	1	14	2	2	1	1	0	1	1	0	0	1	0	0	2	0	0	0	3	3	0	0	37
3	1	0	1	0	0	0	1	2	0	0	1	0	0	2	0	0	0	0	0	1	0	0	0	1	2	0	12
4	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	1	0	5
5	0	0	0	0	0	3	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	2	0	0	0	0	8
6	4	0	0	0	2	9	2	1	0	1	0	0	1	4	0	0	1	0	0	5	0	0	0	2	4	2	38
7	2	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	2	1	0	9
8	0	0	0	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0	2	3	0	0	0	2	1	0	12
9	3	2	1	0	0	2	2	1	1	0	1	1	0	0	0	0	0	0	2	2	2	0	1	2	4	0	27
10	1	0	0	0	1	14	1	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	21
11	0	2	0	1	1	0	4	1	1	3	9	4	5	1	1	1	3	9	0	0	0	1	3	1	0	1	52
12	1	25	12	22	8	2	10	11	16	8	4	9	10	10	9	3	10	6	6	3	1	0	0	0	1	1	188
13	0	1	4	2	10	0	4	4	3	4	3	2	9	0	4	5	2	8	10	6	1	1	1	3	0	2	89
14	1	0	7	3	3	0	3	3	4	1	6	4	1	7	7	8	11	2	3	1	1	0	1	0	1	2	80
15	0	2	1	0	3	0	1	0	2	2	0	4	1	0	1	1	0	2	1	1	0	0	0	1	0	1	24
16	3	0	0	0	0	0	0	1	0	0	1	4	3	1	1	0	0	2	0	3	1	6	12	5	1	4	48
17	17	7	1	9	3	7	4	1	5	4	5	9	6	9	0	2	6	5	6	5	14	15	14	3	11	14	182
18	1	3	6	4	6	1	11	9	4	3	6	5	1	7	11	10	2	0	7	1	1	0	0	1	3	1	104
19	0	1	1	1	2	0	0	1	0	0	2	1	2	2	0	1	1	2	0	1	3	6	0	4	1	1	33
20	5	0	2	0	0	0	1	3	0	3	0	2	2	1	3	2	0	4	6	8	10	11	14	11	12	6	106
21	10	3	8	6	8	4	3	4	5	12	1	6	6	6	8	5	12	3	5	8	14	6	2	0	13	17	175
22	1	7	11	5	8	0	6	7	10	9	2	3	2	4	5	15	7	2	2	2	4	0	0	3	0	1	116
23	1	0	0	0	0	0	0	1	2	1	3	1	4	1	0	0	1	7	0	0	1	1	3	0	0	2	29
24	1	0	0	1	0	0	2	2	2	2	1	1	2	0	0	0	0	4	2	1	1	3	0	1	0	0	26
25	3	2	1	2	1	0	1	2	0	3	2	0	2	0	6	4	1	1	3	7	3	7	4	6	2	4	67
26	4	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	2	1	2	9	2	1	24

Table 37. Confusion matrix for cell-based localization experiment using O1 data set

Predicted Cell of Target	Actual Cell of Target																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
1	0	0	6	2	3	0	3	8	6	7	3	2	4	3	7	3	7	4	0	2	8	5	0	0	1	8	92
2	2	0	1	0	1	0	0	1	0	0	0	1	3	0	0	0	0	2	2	3	3	6	5	3	3	4	40
3	0	7	14	11	18	0	10	16	11	4	22	13	15	9	16	20	6	15	15	5	2	0	0	0	0	2	231
4	5	1	1	0	2	0	2	2	0	2	0	0	0	0	3	1	0	1	5	12	5	0	4	5	4	2	57
5	2	3	0	0	1	1	3	2	1	1	0	3	1	2	1	1	2	2	3	7	2	4	2	3	0	2	49
6	2	0	4	0	1	2	0	1	1	0	0	0	0	0	2	0	0	0	1	3	0	1	1	14	4	1	38
7	2	18	13	19	15	1	17	8	14	12	9	12	19	8	9	10	8	14	5	4	3	2	2	0	1	6	231
8	3	2	2	2	3	0	4	5	4	1	8	4	5	8	5	4	11	5	6	6	5	4	1	2	2	3	105
9	2	2	2	1	3	1	2	3	0	2	0	1	0	0	2	3	3	0	0	2	2	3	2	1	1	2	40
10	6	3	3	5	4	2	1	1	0	2	6	2	2	4	1	2	1	1	4	4	4	3	2	2	9	8	82
11	5	0	0	0	1	1	3	0	1	1	3	3	0	1	0	0	2	1	0	2	4	3	18	1	1	1	52
12	16	14	6	16	2	12	5	4	8	15	0	12	5	15	1	3	13	8	5	1	13	16	16	3	7	16	232
13	0	0	0	1	0	1	0	0	0	3	0	0	2	0	3	5	0	0	1	2	1	1	2	1	2	1	26
14	2	4	0	1	2	0	1	1	3	4	2	2	3	4	3	2	4	7	4	2	1	6	0	0	0	0	58
15	8	1	0	0	0	1	1	0	0	1	1	0	0	0	1	0	0	0	2	1	4	3	0	8	21	3	56
16	1	1	0	0	0	11	1	0	1	1	1	1	0	0	0	3	0	0	1	0	1	0	1	1	0	0	25
17	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2
18	0	0	0	0	0	4	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	2	0	0	11
19	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	6
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
21	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	2
22	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	4
23	0	3	7	2	1	18	3	1	9	1	0	1	0	3	4	2	1	0	2	2	0	1	1	1	2	1	66
24	0	0	0	0	1	0	1	2	0	1	0	1	1	0	0	0	0	0	0	1	1	1	0	3	0	0	13
25	1	0	0	0	1	2	1	4	1	1	2	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	16
26	2	0	1	0	1	1	1	0	0	1	1	0	0	1	2	1	1	0	2	0	0	0	1	8	1	0	25

Table 38. Confusion matrix for cell-based localization experiment using O2 data set

Predicted Cell of Target	Actual Cell of Target																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
1	1	13	19	8	12	0	17	19	21	16	23	6	13	11	19	19	12	12	8	3	2	6	3	1	0	1	265
2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	2	0	6
3	8	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	1	1	0	0	2	1	3	2	5	0	27
4	0	1	5	0	2	9	2	1	4	3	2	5	2	0	1	4	1	0	3	2	9	4	3	5	1	1	70
5	2	0	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0	1	0	2	6	0	14
6	0	3	0	2	1	0	1	0	0	0	4	0	0	0	1	1	0	1	1	2	1	0	0	1	1	0	20
7	14	1	1	1	0	0	0	0	0	2	1	1	0	1	0	0	0	1	5	6	8	11	21	6	12	12	104
8	0	0	0	0	1	3	0	0	0	1	0	0	0	1	0	0	0	0	2	1	0	0	0	0	1	0	10
9	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	6
10	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	5
11	14	27	17	36	12	2	18	12	23	23	15	40	40	40	18	20	37	41	11	4	18	29	5	1	3	24	530
12	0	0	0	0	0	10	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	13
13	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
14	1	1	0	1	0	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	9
15	1	1	3	3	2	1	0	4	0	0	0	2	0	0	2	0	0	0	4	4	1	0	0	1	0	0	29
16	0	0	0	0	0	3	2	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	8
17	4	7	3	0	19	1	9	10	6	4	0	3	3	1	9	10	2	0	6	14	8	0	0	7	8	9	143
18	3	5	8	4	5	9	4	1	1	5	0	1	1	1	4	4	2	0	10	11	4	1	0	6	4	2	96
19	0	0	0	1	0	3	1	0	1	0	0	1	0	0	1	0	0	0	1	2	0	0	0	0	0	1	12
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	2
21	3	0	0	0	1	4	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	2	1	0	13
22	1	0	1	1	1	3	1	5	1	0	1	0	0	2	2	0	0	0	4	4	0	0	0	2	2	1	32
23	1	1	0	0	2	1	2	0	1	1	6	1	0	1	0	0	2	4	0	0	1	4	21	1	2	1	53
24	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
25	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2
26	6	0	3	1	2	0	2	6	1	5	2	0	1	0	1	1	1	0	4	4	5	1	2	20	10	7	85

Table 39. Confusion matrix for cell-based localization experiment using O3 data set

Predicted Cell of Target	Actual Cell of Target																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
1	2	14	19	21	22	0	13	15	18	7	33	24	22	16	19	26	18	28	17	6	3	0	1	3	1	7	355
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	2	0	6
3	6	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	4	5	2	15	5	9	2	53
4	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	3	0	0	8
5	0	0	0	0	0	8	0	0	0	0	2	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	13
6	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	9
7	7	1	2	7	10	0	6	4	4	8	4	5	22	6	7	6	6	13	6	4	4	19	7	1	0	8	167
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
11	10	11	1	10	2	9	5	1	6	13	0	11	5	18	3	2	15	7	1	1	10	7	0	0	4	10	162
12	13	12	9	11	4	1	7	8	1	9	7	7	7	6	8	4	4	5	16	24	18	25	20	18	17	19	280
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	8	12	20	7	9	9	8	13	19	15	0	6	1	5	13	7	10	1	7	7	14	2	2	4	9	10	218
18	0	0	0	0	0	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
19	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
20	0	0	1	0	0	1	2	1	0	1	2	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	10
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	6	10	7	4	10	18	14	13	8	4	0	3	1	7	10	14	3	0	9	10	3	0	0	4	9	2	169
23	1	0	0	0	2	2	3	0	3	1	7	2	1	1	0	0	2	4	0	0	0	3	11	1	1	0	45
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	5	0	1	0	1	0	1	5	1	1	3	0	0	0	0	0	0	0	2	3	1	0	1	20	8	1	54

Table 40. Confusion matrix for cell-based localization experiment using O4 data set

Predicted Cell of Target \ Actual Cell of Target	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
1	3	17	26	31	17	0	25	20	32	25	22	33	39	34	29	32	36	32	16	11	8	12	1	0	0	12	513
2	0	0	0	1	0	2	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	0	3	0	0	10
3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
6	0	0	0	1	1	6	1	3	2	0	0	0	0	3	1	0	1	0	0	2	0	1	1	0	0	0	23
7	13	2	4	2	3	0	2	3	0	6	2	8	8	5	4	1	5	10	4	8	16	22	21	14	12	13	188
8	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4
9	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	3
10	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
11	0	1	0	1	1	4	2	1	1	1	18	0	1	2	0	0	1	10	0	0	0	2	8	1	1	1	57
12	22	14	7	9	14	0	8	13	6	10	2	11	6	7	12	8	4	4	23	23	18	16	14	19	29	17	316
13	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
14	3	2	1	0	1	1	0	2	0	4	1	1	0	1	1	1	2	0	1	2	1	2	2	3	2	4	38
15	0	0	0	0	0	3	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	6
16	1	0	0	0	0	4	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	8
17	4	20	16	11	18	14	16	10	14	7	0	1	3	6	8	9	7	0	8	6	8	0	0	0	8	8	202
18	1	1	1	1	0	1	1	2	1	1	0	0	0	0	2	2	0	0	2	0	0	0	0	2	0	0	18
19	0	0	0	0	0	3	0	0	1	0	0	2	1	0	0	1	1	0	0	0	1	2	0	2	0	0	14
20	0	0	0	1	0	3	0	1	1	0	0	1	0	0	0	1	0	0	1	1	2	0	0	0	0	0	12
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	3
22	1	1	4	1	2	1	1	1	1	0	0	0	0	0	2	3	1	0	2	3	1	0	0	0	0	1	26
23	0	1	0	0	2	4	3	1	1	0	14	2	2	1	1	0	2	4	0	1	2	3	11	0	1	0	56
24	2	0	0	0	1	6	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	12
25	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	5
26	8	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	3	0	1	14	7	2	39

Table 41. Confusion matrix for cell-based localization experiment using OAvg data set

Table 42. Context-based localization results using threshold requirements (g)

		$g=0$			$g=1$			$g=2$			$g=3$			$g=4$			$g=5$		
Trial	True X	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	1	2	1	1	2	1	1	2	1	1	2	1	1	2	1	1	2	1	2
2	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0
3	1	1	1	3	1	1	3	1	1	3	1	1	3	0	1	3	0	1	0
4	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	1
5	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	2	0	0	0
6	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	0	0	1
7	2	1	3	1	1	3	1	1	3	1	1	3	1	1	3	1	1	0	0
8	2	3	2	1	3	2	1	3	2	1	3	1	1	3	1	1	0	0	0
9	2	2	1	1	2	1	1	2	1	1	2	1	1	2	1	1	0	1	1
10	2	1	2	1	1	2	1	1	2	1	1	2	1	0	2	1	0	2	0
11	3	3	3	1	3	3	1	3	3	1	3	3	1	3	3	1	2	3	0
12	3	2	2	3	2	2	3	2	2	3	2	2	3	0	0	3	0	0	0
13	3	3	3	3	3	3	3	3	3	3	3	3	3	0	3	3	0	3	0
14	3	2	3	2	2	3	2	2	3	2	2	3	2	2	0	1	0	0	0
15	3	3	0	3	3	0	3	3	0	3	3	0	0	3	0	0	3	0	0
16	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
17	0	3	2	3	3	2	3	3	2	3	3	2	1	0	2	0	0	0	0
18	0	1	1	2	1	1	2	1	1	2	1	1	2	1	0	2	0	0	0
19	0	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	1	1	1
20	0	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1

Table 43. Context-based localization results using threshold requirements (g) in Room B

		$g=0$			$g=1$			$g=2$			$g=3$			$g=4$			$g=5$		
Trial	True X	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	1	1	2	2	1	2	2	1	2	2	1	2	2	1	0	0	0	0	0
2	1	2	1	2	2	1	2	2	1	2	2	1	2	1	1	2	0	0	0
3	1	2	1	2	2	1	2	2	1	2	2	1	2	2	1	2	2	0	0
4	1	1	2	2	1	2	2	1	2	2	1	2	2	1	2	2	1	2	0
5	1	2	1	1	2	1	1	2	1	1	2	1	1	2	1	1	0	1	1
6	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	1	0	0
7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	2	0	0	0
8	2	1	2	2	1	2	2	1	2	2	1	2	2	1	2	2	0	0	2
9	2	2	2	1	2	2	1	2	2	1	2	2	1	0	2	1	0	0	0
10	2	1	1	2	1	1	2	1	1	2	0	0	0	0	0	0	0	0	0
11	0	2	1	2	2	1	2	2	1	2	2	1	2	0	1	2	0	0	0
12	0	1	1	2	1	1	2	1	1	2	1	2	2	0	0	0	0	0	0
13	0	1	1	2	1	1	2	1	1	2	1	1	2	1	0	0	0	0	0
14	0	1	1	2	1	1	2	1	1	2	2	0	2	2	0	0	0	0	0
15	0	1	2	1	1	2	1	1	2	1	1	2	2	0	2	0	0	0	0

Table 44. Context-based localization results using threshold requirements (g) in Room C

		$g=0$			$g=1$			$g=2$			$g=3$			$g=4$			$g=5$		
Trial	True X	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	1	2	1	1	2	1	1	2	1	1	2	1	1	2	1	1	2	0	3
2	1	3	3	2	3	3	2	3	3	2	3	3	2	3	3	2	3	3	2
3	1	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	3	2	0
4	1	2	1	3	2	1	3	2	1	3	2	1	3	2	1	3	0	0	0
5	1	3	1	3	3	1	3	3	1	3	3	1	3	3	0	2	0	0	0
6	2	3	2	3	3	2	3	3	2	3	3	2	3	3	2	3	0	0	0
7	2	3	2	3	3	2	3	3	2	3	3	2	3	3	2	3	2	0	0
8	2	3	2	2	3	2	2	3	2	2	3	2	2	3	2	2	2	2	2
9	2	2	2	3	2	2	3	2	2	3	2	0	3	2	0	3	0	0	0
10	2	2	2	3	2	2	3	2	2	3	2	2	3	2	2	3	0	2	3
11	3	2	2	1	2	2	1	2	2	1	2	2	1	2	2	1	0	2	2
12	3	1	3	2	1	3	2	1	3	2	1	3	2	0	0	3	0	0	0
13	3	2	1	2	2	1	2	2	2	2	2	2	2	2	0	3	0	0	0
14	3	2	2	3	2	2	3	2	2	3	2	2	3	2	2	3	2	2	0
15	3	3	3	1	3	3	1	3	3	1	3	3	1	3	3	1	3	3	0
16	0	3	3	2	3	3	2	3	3	2	3	3	2	3	3	1	0	0	0
17	0	1	3	3	1	3	3	1	3	3	1	3	3	1	3	3	2	2	3
18	0	2	3	2	2	3	2	2	3	2	2	3	2	2	3	2	2	0	0
19	0	2	3	3	2	3	3	2	3	3	2	3	3	2	0	3	3	0	3
20	0	2	3	2	2	3	2	2	3	2	2	3	2	2	3	2	0	3	2

Bibliography

1. N. Pirzada, M. Y. Nayan, F. S. M. F. Hassan, and M. A. Khan, "Device-free Localization Technique for Indoor Detection and Tracking of Human Body: A Survey," in *Procedia - Social and Behavioral Sciences*, vol. 129, May 2014, pp. 422–429.

2. A. Paul and T. Sato, "Localization in Wireless Sensor Networks: A Survey on Algorithms, Measurement Techniques, Applications and Challenges," *Journal of Sensor and Actuator Networks*, vol. 6, no. 4, p. 24, Oct 2017.

3. J. Xiao, Z. Zhou, Y. Yi, and L. M. Ni, "A Survey on Wireless Indoor Localization from the Device Perspective," *ACM Computing Surveys*, vol. 49, no. 2, Nov 2016.

4. W. Ruan, Q. Z. Sheng, L. Yao, T. Gu, M. Ruta, and L. Shangguan, "Device-free indoor localization and tracking through Human-Object Interactions," in *17th International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Jun 2016.

5. N. Patwari and J. Wilson, "RF Sensor Networks for Device-Free Localization: Measurements, Models, and Algorithms," in *Proceedings of the IEEE*, vol. 98, no. 11, Jul 2010, pp. 1961–1973.

6. L. Cheng, C. Wu, Y. Zhang, H. Wu, M. Li, and C. Maple, "A survey of localization in wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 2012, Dec 2012.

7. F. Viani, F. Robol, A. Polo, P. Rocca, G. Oliveri, and A. Massa, "Wireless architectures for heterogeneous sensing in smart home applications: Concepts and real implementation," in *Proceedings of the IEEE*, vol. 101, no. 11, Jul 2013, pp. 2381–2396.

8. G. Deak, K. Curran, J. Condell, and D. Deak, "Motion Detection using Device-free Passive Localisation (DfPL)," in *International Signals and Systems Conference*, Jun 2012.

9. C. Xu, B. Firner, Y. Zhang, and R. E. Howard, "The Case for Efficient and Robust RF-Based Device-Free Localization," *IEEE Transactions on Mobile Computing*, vol. 15, no. 9, pp. 2362–2375, Oct 2016.

10. M. Youssef, M. Mah, and A. Agrawala, "Challenges: Device-free Passive Localization for Wireless Environments," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, Jan 2007, p. 222.

11. S. Shukri, L. M. Kamarudin, G. C. Cheik, R. Gunasagaran, A. Zakaria, K. Kamarudin, S. M. Zakaria, A. Harun, and S. N. Azemi, "Analysis of RSSI-based DFL for Human Detection in Indoor Environment using IRIS mote," in *3rd International Conference on Electronic Design*, Aug 2016, pp. 216–221.

12. S. Hara and D. Anzai, “Experimental Performance Comparison of RSSI- and TDOA-Based Location Estimation Methods,” in *IEEE Vehicular Technology Conference*, May 2008, pp. 2651–2655.

13. P. Vorst, J. Sommer, C. Hoene, P. Schneider, C. Weiss, T. Schairer, W. Rosenstiel, A. Zell, and G. Carle, “Indoor Positioning via Three Different RF Technologies,” in *4th European Workshop on RFID Systems and Technologies*, Jun 2008.

14. N. Kuxdorf-Alkirata and D. Brückmann, “Reliable and Low-cost Indoor Localization Based on Bluetooth Low Energy,” in *The 3rd IEEE International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems*, Sep 2016, pp. 92–96.

15. J. Kemper and H. Linde, “Challenges of passive infrared indoor localization,” in *5th Workshop on Positioning, Navigation and Communication*, Mar 2008, pp. 63–70.

16. S. Sen, J. Lee, K.-h. Kim, and P. Congdon, “Avoiding multipath to revive inbuilding WiFi localization,” in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, Jun 2013, p. 249.

17. A. T. Parameswaran, M. I. Husain, and S. Upadhyaya, “Is RSSI a reliable parameter in sensor localization algorithms - an experimental study,” in *IEEE International Symposium on Reliable Distributed Systems*, Jan 2009.

18. A. E. Kosba, A. Abdelkader, and M. Youssef, “Analysis of a Device-Free Passive Tracking System in Typical Wireless Environments,” in *3rd International Conference on New Technologies, Mobility and Security*, Dec 2009.

19. G. Deak, K. Curran, J. Condell, E. Asimakopoulou, and N. Bessis, “IoTs (Internet of Things) and DfPL (Device-free Passive Localisation) in a disaster management scenario,” *Simulation Modelling Practice and Theory*, vol. 35, pp. 86–96, Jun 2013.

20. A. Narzullaev, Y. Park, and H. Jung, “Accurate signal strength prediction based positioning for indoor WLAN systems,” in *Record - IEEE PLANS, Position Location and Navigation Symposium*, May 2008, pp. 685–688.

21. C. Xu, “Device-Free People Counting and Localization,” in *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing adjunct publication*, Sep 2013, pp. 367–372.

22. M. Seifeldin and M. Youssef, “A Deterministic Large-Scale Device-Free Passive Localization System for Wireless Environments,” in *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, Jun 2010.

23. O. Kaltiokallio, M. Bocca, and N. Patwari, “Follow @grandma: Long-term device-free localization for residential monitoring,” in *Proceedings - Conference on Local Computer Networks*, Oct 2012, pp. 991–998.

24. P. Cassarà, F. Potortì, P. Barsocchi, and M. Girolami, "Choosing an RSS device-free localization algorithm for Ambient Assisted Living," in *International Conference on Indoor Positioning and Indoor Navigation*, Oct 2015.

25. H. Clougherty, A. Brown, M. Stonerock, M. Trepte, M. Whitesell, and R. Bailey, "Home automation and personalization through individual location determination," in *Systems and Information Engineering Design Symposium*, Apr 2017, pp. 300–305.

26. R. Elbakly and M. Youssef, "A Calibration-free RF Localization System," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Nov 2015, pp. 63:1—-63:2.

27. I. Oksar, "A Bluetooth Signal Strength Based Indoor Localization Method," in *International Conference on Systems, Signals and Image Processing*, no. May, May 2014, pp. 251 – 4.

28. C. M. Angelopoulos, G. Filios, M. Karagiannis, S. Nikoletseas, and J. Rolim, "Fine-grained in-door localisation with wireless sensor networks," in *Proceedings of the 10th ACM International Symposium on Mobility Management and Wireless Access*, Oct 2012, pp. 159–162.

29. J. Yang and Y. Chen, "Indoor Localization Using Improved RSS-Based Lateration Methods," in *Global Telecommunications Conference*, 2009.

30. X. Kuang and H. Shao, "Maximum Likelihood Localization Algorithm Using Wireless Sensor Networks," in *First International Conference on Innovative Computing, Information and Control*, vol. I, Aug-Sep 2006, pp. 263–266.

31. A. E. Waadt, C. Kocks, S. Wang, G. H. Bruck, and P. Jung, "Maximum likelihood localization estimation based on received signal strength," in *3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, Nov 2010, pp. 1–5.

32. R. Priwgharm and P. Chemtanomwong, "A Comparative Study on Indoor Localization based on RSSI Measurement in Wireless Sensor Network," in *Eighth International Joint Conference on Computer Science and Software Engineering*, May 2011.

33. G. Zanca, F. Zorzi, A. Zanella, and M. Zorzi, "Experimental comparison of RSSI-based localization algorithms for indoor wireless sensor networks," in *Proceedings of the workshop on Real-world wireless sensor networks*, Apr 2008.

34. P. Cherntanomwong and D. J. Suroso, "Indoor localization system using wireless sensor networks for stationary and moving target," in *8th International Conference on Information, Communications & Signal Processing*, Dec 2011.

35. M. Alkandari, D. Basu, and S. F. Hasan, “A Wi-Fi based passive technique for speed estimation in indoor environments,” in *Proceedings of the 2017 2nd Workshop on Recent Trends in Telecommunications Research*, Feb 2017, pp. 10–12.

36. J. S. Turner, M. F. Ramli, L. M. Kamarudin, A. Zakaria, A. Y. Shakaff, D. L. Ndzi, C. M. Nor, N. Hassan, and S. M. Mamduh, “The Study of Human Movement Effect on Signal Strength for Indoor WSN Deployment,” in *IEEE Conference on Wireless Sensor*, Dec 2013, pp. 30–35.

37. S. H. Ahmed, S. H. Bouk, N. Javaid, and I. Sasase, “Combined Human, Antenna Orientation in Elevation Direction and Ground Effect on RSSI in Wireless Sensor Networks,” in *Proceedings - 10th International Conference on Frontiers of Information Technology*, Dec 2012, pp. 46–49.

38. H. Aly and M. Youssef, “New Insights into Wifi-based Device-free Localization,” in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, Sep 2013, pp. 541–548.

39. ——, “An Analysis of Device-Free and Device-Based WiFi-Localization Systems,” *International Journal of Ambient Computing and Intelligence*, vol. 6, no. 1, pp. 1–19, Jan 2014.

40. J. Wang, X. Zhang, Q. Gao, X. Ma, X. Feng, and H. Wang, “Device-free simultaneous wireless localization & activity recognition with wavelet feature,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1659–1669, Apr 2017.

41. M. Seifeldin and M. Youssef, “Nuzzer : A Large-Scale Device-Free Passive Localization System for Wireless Environments,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1321–1334, May 2013.

42. Y. Zhao and N. Patwari, “Noise reduction for variance-based device-free localization and tracking,” in *8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Jun 2011, pp. 179–187.

43. M. Wang, Z. Wang, X. Bu, and E. Ding, “An adaptive weighting algorithm for accurate radio tomographic image in the environment with multipath and WiFi interference,” *International Journal of Distributed Sensor Networks*, vol. 13, no. 1, Jan 2017.

44. S. Nannuru, Y. Li, Y. Zeng, M. Coates, and B. Yang, “Radio-frequency tomography for passive indoor multitarget tracking,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 12, pp. 2322–2333, Sep 2013.

45. H. Sharma and S. Sharma, “A review of sensor networks: Technologies and applications,” in *Recent Advances in Engineering and Computational Sciences*, Mar 2014, pp. 6–8.

46. INSTEON, “INSTEON WHITEPAPER: Compared ,” INSTEON, Tech. Rep., 2013. [Online]. Available: http://cache.insteon.com/pdf/INSTEONCompared.pdf

47. C. Gomez and J. Paradells, “Wireless home automation networks: A survey of architectures and technologies,” *IEEE Communications Magazine*, vol. 48, no. 6, pp. 92–101, May 2010.

48. W. Lumpkins, “Home Automation: Insteon (X10 Meets Powerline),” *IEEE Consumer Electronics Magazine*, vol. 3, no. October, pp. 140–144, Oct 2015.

49. C. M. Talbot, M. A. Temple, T. J. Carbino, and J. A. Betances, “Detecting rogue attacks on commercial wireless Insteon home automation systems,” *Computers and Security*, vol. 2017, 2017.

50. T. S. P. See, X. Qing, Z. N. Chen, C. K. Goh, and T. M. Chiam, “RF transmission in-/through the human body at 915 MHz,” in *IEEE International Symposium on Antennas and Propagation and CNC-USNC/URSI Radio Science Meeting - Leading the Wave*, Jul 2010.

51. INSTEON, “INSTEON WHITEPAPER: The Details,” INSTEON, Tech. Rep., 2013. [Online]. Available: http://cache.insteon.com/pdf/insteondetails.pdf

52. *INSTEON Developer’s Guide 2nd Edition*, SmartLabs Technology, August 2007. [Online]. Available: http://cache.insteon.com/pdf/INSTEON_Developers_Guide_20070816a.pdf

53. *INSTEON Hub: Developer’s Guide*, SmartLabs Technology, August 2013. [Online]. Available: http://cache.insteon.com/developer/2242-222dev-062013-en.pdf

54. “NooElec NESDR Mini 2 SDR & DVB-T USB Stick (RTL2832 + R820T2) w/ Antenna and Remote Control,” Accessed: 2017-04-30. [Online]. Available: http://www.nooelec.com/store/sdr/sdr-receivers/nesdr-mini2-rtl2832u-r820t2.html

55. “Products: ECCOSORB® AN.” [Online]. Available: http://www.eccosorb.com/products-eccosorb-an.htm

56. Insteon, “Insteon API,” Dec 2016, Accessed: 2017-09-24. [Online]. Available: https://insteon.docs.apiary.io/

57. INSTEON, “Insteon for Hub APK 1.9.8,” Sep 2017, Accessed: 2017-10-28. [Online]. Available: https://apkpure.com/insteon-for-hub/com.insteon.insteon3

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
22-03-2018	Master's Thesis	Jun 2016 - Mar 2018

4. TITLE AND SUBTITLE

RSS-based Device-free Passive Detection and Localization Using Home Automation Network Radio Frequencies

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Phan, Tiffany M., 2d Lt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

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

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-18-M-054

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

Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

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

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution Statement A. Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This research provided a proof of concept for a device-free passive (DfP) system capable of detecting and localizing a target through the exploitation of the Insteon home automation network's 915 MHz radio frequency (RF) signals. Software-defined radios simulated Insteon devices capable of collecting and reporting RSS. A message generation script was created and a calibrated filter was implemented to reduce silent periods. Baseline threshold detection was conducted using a tripwire setup and performed well. Multi-node, cell-based localization was tested and received poor results. A context-based method was developed and tested in the same room as well as two other rooms and observed improved localization performance. Feasibility of exploiting RF of a home automation network for DfP indoor detection and localization was demonstrated. Despite not achieving optimal localization performance, the results from system deployment showed promise for future DfP system deployment on top of home automation RF devices.

15. SUBJECT TERMS

Internet of Things, Home Automation, Device-free Passive, Detection, Localization

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Richard K. Martin, (ENG)
U	U	U	UU	179	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636 x4625 richard.martin@afit.edu