A REAL-TIME LABORATORY TESTBED FOR EVALUATING LOCALIZATION PERFORMANCE OF WIFI RFID TECHNOLOGIES

A Thesis submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical and Computer Engineering

by

____

Muhammad Ali Assad

(maassad@wpi.edu)

May 4th 2007

APPROVED:

______________________________	______________________________
Prof. Kaveh Pahlavan, Thesis Advisor	Prof. Fred J. Looft, Department Head
______________________________	______________________________
Prof. Allen Levesque, Thesis Committee	Prof. Wenjing Lou, Thesis Committee

Abstract

A realistic comparative performance evaluation of indoor Geolocation systems is a complex and challenging problem facing the research community. This is due to the fact that performance of these systems depends on the statistical variations of the fading multipath characteristics of the wireless channel, the density and distribution of the access points in the area, and the number of the training points used by the positioning algorithm. This problem, in particular, becomes more challenging when we address RFID devices, because the RFID tags and the positioning algorithm are implemented in two separate devices. In this thesis, we have designed and implemented a testbed for comparative performance evaluation of RFID devices in a controlled and repeatable laboratory environment. The testbed consists of a real-time RF channel simulator, several WiFi 802.11 access points, commercial RFID tags, and a laptop loaded with the positioning algorithm and its associated user interface. In the real-time channel simulator, the fading multipath characteristics of the wireless channel between the access points and the RFID tags is modeled by a modified site-specific IEEE 802.11 channel model which combines this model with the correlation model of shadow fading existing in the literature. The testbed is first used to compare the performance of the modified IEEE 802.11 channel model and the Ray Tracing channel model previously reported in the literature. Then, the testbed with the new channel model is used for comparative performance evaluation of two different WiFi RFID devices.

Acknowledgements

I would like to express my profound gratitude to my thesis advisor, Dr. Kaveh Pahlavan, whose continuous guidance, support and wisdom helped me focus and attain the objectives of this research.

I would specifically like to thank the members of the M.S. thesis committee, Dr. Allen Levesque and Dr. Wenjing Lou for their valuable time and comments on my research.

I would also like to thank my lab peers, Ferit Akgul, Nayef Alsindi and especially Mohammad Heidari for providing continuous guidance and support over the course of the research. They also motivated me towards achieving my research goals by maintaining a healthy learning environment in the CWINS lab.

I would like to dedicate this work to my grandmother, and my parents whose prayers, love and support was the ultimate motivation during my graduate studies.

Finally, but foremost, I am always indebted to Allah All Mighty, the Most Beneficent and the Most Merciful, for keeping me in a state of well being and for helping me attain all the knowledge and appreciation throughout my studies. ALHAMDULILLAH

Table of Contents

List of Figures

List of Tables

Glossary

AK	Atwater Kent
AOA	Angle of Arrival
API	Application Program Interface
AP	Access Point
CDF	Cumulative Distribution Function
CIR	Channel Impulse Response
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DSSS	Direct Sequence Spread Spectrum
EPE	Ekahau Positioning Engine
FDP	First Detected Peak
GPS	Global Positioning System
ICT	Instrument Control Toolbox
ISO	International Standards Organization
LAN	Local Area Network
LOS	Line of Sight
OOK	On/Off Key
PC	Personal Computer

PDP	Path Delay Profile
RFID	Radio Frequency Identification Device
RMS	Root Mean Square
RMSE	Root Mean Square Error
RP	Reference Point
RSS	Received Signal Strength
RT	Ray Tracing
RTLS	Real-Time Locating System
TDOA	Time Difference of Arrival
TOA	Time of Arrival
TP	Training Point
UDP	Undetected Direct Path
UWB	Ultra-WideBand
XML	eXtensible Markup Language

Chapter 1: Introduction

This chapter is divided into four sections. Section 1.1 provides the background of the localization industry. Section 1.2 gives an overview of the previous work and the motivation behind this thesis. Section 1.3 highlights the major contributions made through this research. Section 1.4 provides an outline of the remaining chapters in this thesis.

1.1 Background of the Industry

Recently, there has been an enormous amount of innovative development of active RFID devices. These RFID devices are being integrated with traditional localization techniques and are becoming an essential part of Real-Time Locating Systems (RTLS). Today many industries are using RTLS for localization and identification of various products and assets in commercial, health care, public safety, and military domains. In the commercial domain, localization has many potential uses in warehousing and in supply-chain management. In the health care domain, there are important uses for tracking/locating patients, medications, and instruments in hospitals, as well as tracking people with special needs. In the public safety and military domains, precise localization can be utilized to assist firefighters and military personnel in accomplishing their missions [1].

There are a variety of localization techniques used by RFID manufacturers. These techniques include Sign Post, Received Signal Strength (RSS) and Time of Arrival (TOA). Sign Post technique has been implemented in conjunction with passive RFID technologies by companies such as Omron. Sign Post requires no battery and uses simple hardware and simple algorithms. However, this technique is not accurate and is not widely used for localization purposes. RSS technique is extensively used for WiFi Positioning by Ekahau, G2 Microsystems, Newbury Networks and SkyHook. This technique uses simple hardware but requires a tedious training procedure and complex algorithms for accurate localization. TOA, on the other hand, needs no training and requires only few reference points for accurate indoor positioning. However, it requires synchronization with reference points and is sensitive to multipath and blockage. TOA localization technique is used by GPS systems, PinPoint, WhereNet and IEEE 802.15.3 [1].

The RTLS solutions available in the market are using RFID devices that are based on various wireless standards. Some of the most commonly used RFID devices are WiFi RFID (IEEE 802.11 a,b,g,n), Ultra Wide Band (UWB) RFID (IEEE 802.15.3) and ZigBee RFID (IEEE 802.15.4). These technologies utilize different power levels, use a variety of PHY and MAC layer techniques, and employ different bandwidths and frequency of operation [10]. The wide variety of RFID based RTLS technologies and the increasing range of applications was the main motive behind localization standards such as ISO 24730. The ISO 24730 standard has defined guidelines that can be used by RTLS technologies in asset management and for compatibility and interoperability of products in the growing RTLS market [2].

The next section provides an overview of previous work found in the literature with regards to performance evaluation of localization techniques and testbed development for comparative evaluation of localization/positioning systems. It also emphasizes the need of developing a testbed in a controlled and repeatable environment that can be used to evaluate the localization performance of RFID devices.

1.2 Background and Motivation of Research

The coexistence of multiple RFID devices in an integrated system brings unique challenges to both the supply chain and the end users. First, the desire for interoperability among different RFID devices motivates the design of a universal reader that can automatically self-reconfigure to access all types of RFID devices. Second, it is important to understand the interference among different types of RFID devices in real-time operating environments. Third, the capability of RFID technology itself needs to be assessed. Studies [3] conducted by the RFID Alliance Lab found inconsistencies in quality and performance of RFID tags. They found that there are considerable variations in performance from one tag to the other, even among tags of the same manufacturer and model.

This emphasizes the need of a performance benchmark that can aid the growth of the RFID industry and provide support for innovation. A variety of testbeds with different level of complexity are needed to perform comparative evaluations on different RFID technologies. In this research, the design of a real-time channel simulator testbed in a laboratory environment is presented which can be used for evaluating localization performance of WiFi RFID technologies.

Most of the results of performance evaluation of WiFi localization techniques and algorithms presented in the literature were achieved either through on-site measurements or by statistical channel modeling techniques. In [4], a statistical analysis of these models was presented for indoor positioning algorithms. The performance evaluation results were generated using on-site power measurements and computer simulations of channel modeling techniques. In [5], the performance evaluation of localization techniques was conducted by taking real-time measurements through war driving on the Dartmouth university campus. In [6], indoor and outdoor testbed setups were presented for performance evaluation of a localization algorithm. These setups also involved actual measurements of received power inside the building for indoor setup and around the campus buildings for outdoor setup. The site-specific measurements and the war driving efforts, however, do not present a controlled and repeatable test setup that can be used for evaluating comparative performance of WiFi localization technologies such as active RFID tags. Also, since the wireless channel keeps on changing, it is almost impossible to maintain a controlled environment in which real-time measurements can be taken to compare various RFID technologies. Hence, the test scenario involving real-time measurements does not guarantee the same variables for a different set of measurements. Also, it is very time consuming to lead the war driving efforts for recording received powers at different location inside and outside the buildings.

Previous works on a testbed design have been focused mostly on outdoor applications. A verification platform for cellular geo-location systems was presented in [7]. Different setups for simulating real-time radio channels were also studied in [8] but did not focus on indoor areas. The only previous work that can be associated with this thesis was presented in [9]; a real-time testbed design for comparative performance evaluation of indoor geolocation systems was presented. In [9], 2-D Ray Tracing (RT) channel model was used to simulate the channel between the access points and the positioning software.

1.3 Contributions of the Thesis

In this thesis, a real-time testbed design is presented for comparing and evaluating the performance of WiFi RFID technologies with an emphasis on WiFi RFID tag. For channel simulation, site-specific IEEE 802.11 channel model is used, instead of the RT model, which does not require an accurate floor plan of the building and can be implemented without using a third party software.

The contributions of this research are:

1. Developed a site-specific IEEE 802.11 channel model by adding spatially correlated log-normal shadow fading described in the literature to the IEEE 802.11 statistical channel model. Compared the RSS data generated using the site-specific channel model with the RSS data generated using the two-dimensional ray tracing channel model presented earlier in the literature and verified that the data generated by the two channel models is closely correlated.

2. Designed and implemented a testbed that can be used to evaluate the real-time localization performance of active RFID tags in a controlled and repeatable laboratory environment. The testbed is designed using actual wireless hardware including wireless access points, real-time channel emulator hardware, active

RFID modules and a laptop with a PCMCIA wireless card, connected using adaptor cables. The channel model between the wireless access points and the active RFID module is simulated using the site-specific IEEE 802.11 channel model developed during this research.

3. Evaluated the localization performance of the site-specific IEEE 802.11 channel model and the ray tracing channel model. Also, using the site-specific IEEE 802.11 channel model, the localization performance of two different WiFi RFID tags was also evaluated.

1.4 Thesis Outline

This thesis is divided into five main chapters. The first chapter, Chapter 1, provides an overview of the thesis.

In Chapter 2, an overview of the channel modeling principles and techniques is provided. The IEEE 802.11 channel model and the Ray Tracing channel model are described and the pros and cons of these channel models are discussed in regards to indoor localization.

In Chapter 3, a summary of the different localization techniques is provided. In particular, a detailed description of the RSS localization technique is included. In addition, an overview of the emerging RFID technologies is also provided. The importance of RFID technologies for Real-time Location Systems (RTLS) is also discussed.

In Chapter 4, a detailed description of the design of the real-time performance evaluation testbed is provided. The explanation of the different elements found in the testbed and the manner in which they interface with each other are depicted. The technical details required for configuring the different components of the testbed is also provided in the Appendices.

In Chapter 5, two main performance evaluation results are included. The first part of this chapter provides the results of the localization performance of two different channel models. The second section evaluates the localization performance of two different RFID tags developed by the same manufacturer.

In Chapter 6, the conclusions drawn from this thesis are summarized. In addition, some future work is also proposed.

Chapter 2: Channel Models for Indoor Localization

In this chapter, an overview of the channel modeling concepts for indoor environment is provided. In particular, the statistical IEEE 802.11 channel model proposed by the IEEE standardized committee and the building-specific Ray Tracing channel models are compared. These channel models are used in this research for simulating the channel between the access points and the RFID tags. The performance of these channel models for indoor localization is analyzed in Chapter 5.

2.1 Introduction

An accurate characterization of the channel is required for an effective design and assessment of a radio network. The channel characteristics vary from one environment to another, and are used as a basis to determine the advantages of a particular communication technique over another. An accurate channel characterization requires a complete knowledge of key parameters and a detailed mathematical model of the channel. These channel models are an important metrics used to determine the optimum location for installation of antennas and to analyze interference between different systems [10]. There have been many channel models that have been developed and presented by the technical community, standardization committees and researchers for different radio environments. Hence, the choice of an appropriate channel model is very important when evaluating the performance of a particular wireless environment.

Radio propagation in indoor environments is complicated by the fact that the shortest direct path between the transmitter and the receiver is usually blocked by the walls, ceilings or other objects in an interior space. Hence, the signal power is distributed over a multiplicity of paths of various strengths between the transmitter and the receiver. The length of the paths is dependent on the architecture of the environment and the locations of the objects found in the vicinity of the transmitter and the receiver. The arrival times of signals on various paths are proportional to the lengths of the paths. The strengths of the signals on such paths depend on the attenuation caused by transmission through or reflection of the signal by various objects in the path. The deterministic analysis of propagation mechanism is limited to simpler cases, such as a typical indoor office environment. For more complex cases, such as outdoor environments, statistical analysis is more useful and widely used. In statistical modeling, the statistics of the channel parameters are collected from actual measurements at various locations between the transmitter and the receiver [10].

In this thesis, two different channel models have been used. The statistical time-domain channel model proposed by the IEEE 802.11 standardization committee is one of them. The other channel model is the building-specific ray tracing channel model which is more sensitive to the architecture of the building. Section 2.2 describes the ray tracing channel model. In section 2.3, a detailed description of the IEEE 802.11 channel model is provided. In section 2.4, the correlation model for shadow fading is presented. Section 2.5 describes the site-specific IEEE 802.11 channel model developed during this research. Section 2.6 provides a comparison of the IEEE 802.11 channel model and the RT channel model.

2.2 Building Specific Ray Tracing Channel Model

Statistical channel models such as the one mentioned in section 2.3 cannot relate the radio propagation characteristics to the exact locations of the transmitter and the receiver; rather, they provide a collection of possible channel profiles. On the other hand, deterministic radio propagation modeling relates the radio propagation to the physical layout of the building. Ray tracing algorithm, similar to the one mentioned in [11], can provide a relatively approximate solution to indoor radio propagation. In this method, walls, ceilings, and floors are assumed to be dark mirrors. The path between a transmitter and receiver are determined through transmission, reflection and diffraction mechanisms [10]. “For a pair of transmitter and receiver at some known location, RT determines the necessary information of the channel such as the arrival angle, departure angle, phase, number of reflections, and number of transmission by sending a set of rays from the transmitter and tracing them until they either reach the receiver or get largely attenuated that cannot be detected by the receiver [12].”

In this research, RT channel models were generated using PlaceTool. PlaceTool is a CWINS internal software that uses a 2-D ray tracing algorithm to model radio propagation in a typical indoor environment used for wireless LAN applications. As discussed above, this software models all probable paths between the transmitter and the receiver based on the reflections and transmissions through walls and barriers using the information provided in the floor plan. The results include time of arrival, magnitude, phase, and direction of arrival information for each path.

The results generated from the ray tracing are particularly accurate for propagation of radio signals, especially in indoor environments, at frequencies higher than 900 MHz. At these frequencies, the electromagnetic waves can be described as traveling along localized ray paths. RT can be used to produce large databases of channel impulse responses for statistical analysis of the channel and is a possible alternative to physical measurements [10, 12]. Figure 1 shows a typical floor plan with the possible ray paths between the transmitter-receiver pair located at the third floor of Atwater Kent Laboratories in Worcester Polytechnic Institute. Figure 2 represents an RT generated channel impulse response.

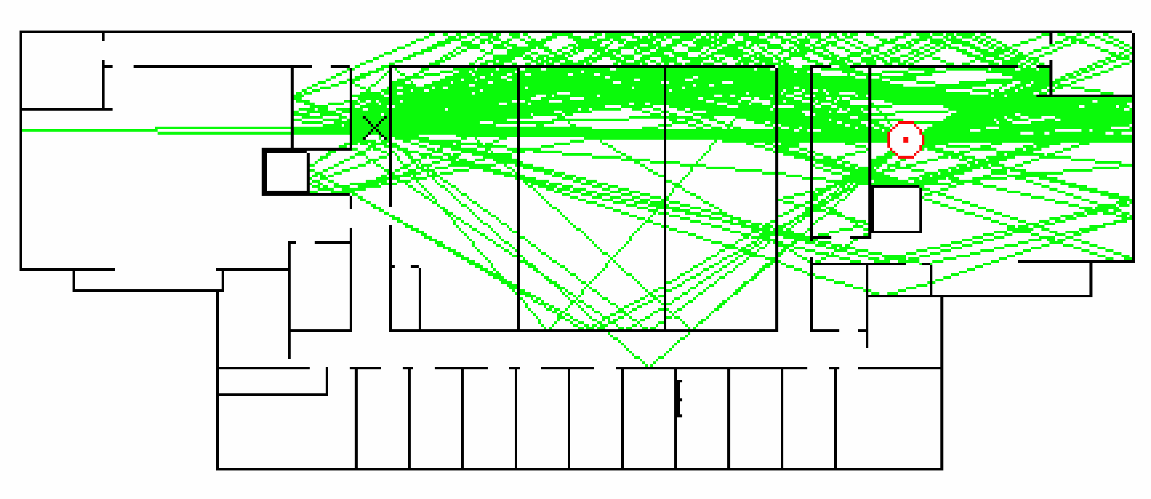


Figure 1: Typical floor plan used for RT channel models

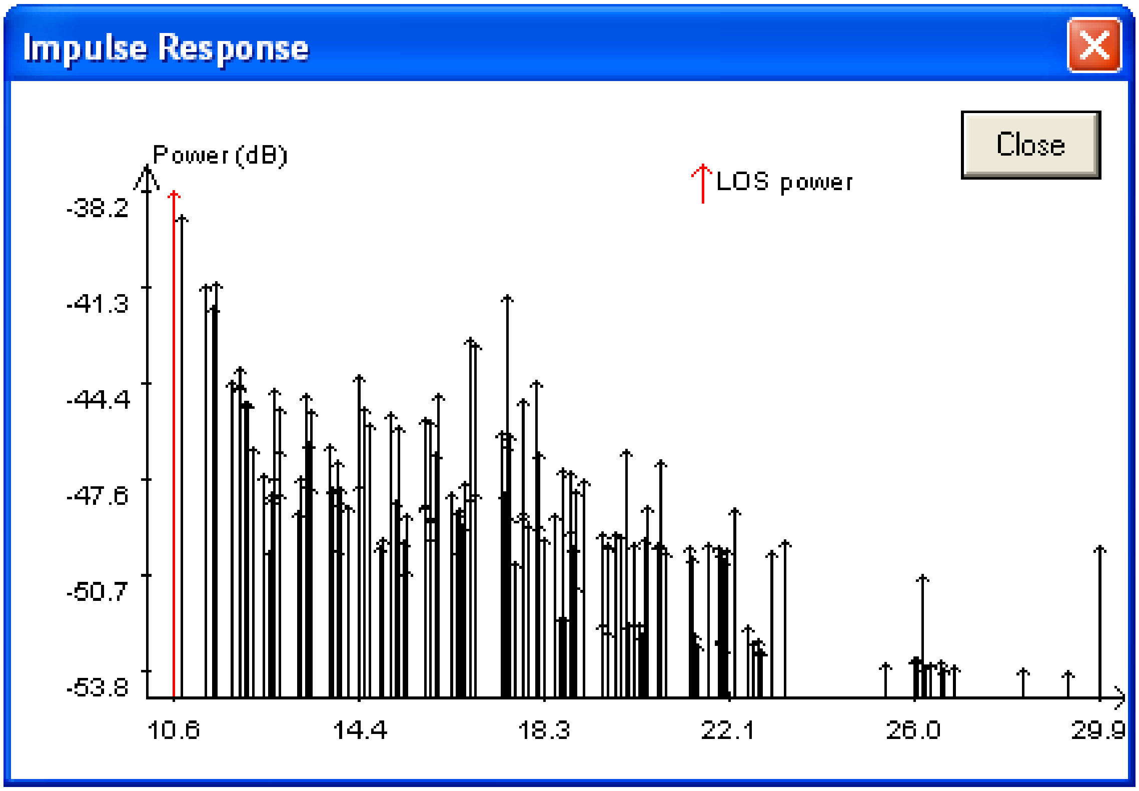
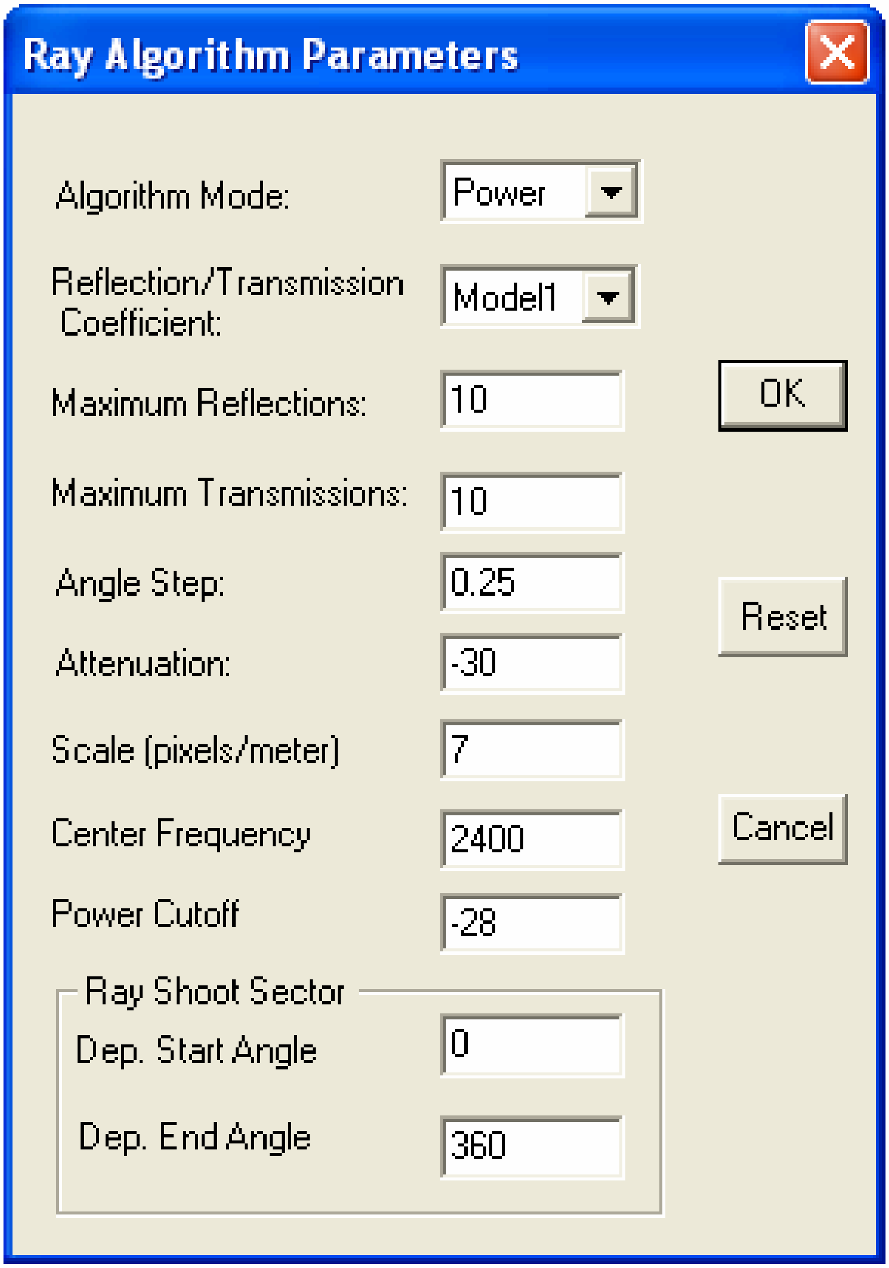


Figure 2: Channel impulse response generated by RT software

Figure 3 shows the configuration of RT parameters used in this research for generating the channel simulations. The complete description of all RT parameters is documented in [13].



Ray Algorithm Parameters

Algorithm Mode: Power

Reflection/Transmission Coefficient: Model1

Maximum Reflections: 10

Maximum Transmissions: 10

Angle Step: 0.25

Attenuation: -30

Scale (pixels/meter) 7

Center Frequency 2400

Power Cutoff -28

Ray Shoot Sector

Dep. Start Angle 0

Dep. End Angle 360

Figure 3: Configuration of RT parameters

The ray tracing software generates a .RAW file that contains the time of arrival, magnitude, phase, and direction of arrival information for each ray path between the transmitter and the receiver. To use the results provided by the RT software, the .RAW file is converted into a .TAP file which is compatible with the hardware emulator used in the testbed. The .TAP file provides the Path Delay Profile (PDP) for a particular set of transmitter and receiver location. The MATLAB code written to transform the .RAW file to .TAP file is provided in Appendix B.6. In addition to the PDP information, the RSS and the distance between the transmitter and receiver are also extracted to implement the shadowing effects. More details on the setup of the hardware emulator used in the testbed are provided in Chapter 4. The MATLAB functions used to extract the distance and RSS information from the .RAW file are provided in Appendices B.2 and B.3.

The impulse response generated by RT software can be represented by Equation 1:

$$h(\tau,t) = \sum_{i=1}^{L} \beta_i e^{j\phi_i} \delta(t-\tau_i) \tag{1}$$

where β_i represents the amplitude of the ith path, φ_i represents the phase of the ith path, τ_i represents the time taken by the ith path to reach the receiver, and L represents the number of ray paths simulated by RT software between the transmitter and the receiver

The RSS information can be extracted from the impulse response using Equation 2:

$$RSS\ (in\ dB) = 10 \cdot \log_{10}\left[\sum_{i=1}^{L} |\beta_i|^2\right] \tag{2}$$

2.3 Statistical Time Domain IEEE 802.11 Channel Model

IEEE 802.11 channel models were originally based on the HiperLAN/2 channel models presented in [14, 15]. Different wireless environments were grouped into five different models (A-E) based on various values of rms delay spread:

- Model A for a typical office environment, Non-Line-Of-Sight (NLOS) conditions, and 50 ns rms delay spread.
- Model B for a typical large open space and office environments, NLOS conditions, and 100 ns rms delay spread.
- Model C for a large open space (indoor and outdoor), NLOS conditions, and 150 ns rms delay spread.
- Model D, same as model C, line-of-sight (LOS) conditions, and 140 ns rms delay spread (10 dB Ricean *K*-factor at the first delay).
- Model E for a typical large open space (indoor and outdoor), NLOS conditions, and 250 ns rms delay spread.

They were then updated to provide a better representation of smaller environments, such as residential homes and small offices. These updates were reported in [16] in which channel coefficients (tap delays and corresponding powers) for six different environments (A-F) were also provided. The resulting models were presented as follows:

- Model A (optional, should not be used for system performance comparisons), flat fading model with 0 ns rms delay spread (one tap at 0 ns delay model). This model can be used for stressing system performance, occurs small percentage of time (locations).

- Model B with 15 ns rms delay spread.

- Model C with 30 ns rms delay spread.

- Model D with 50 ns rms delay spread.

- Model E with 100 ns rms delay spread.

- Model F with 150 ns rms delay spread.

In addition, a path loss model for each environment was also suggested. The path loss model is represented by Equation 3:

$$
\begin{aligned}
L(d) &= L_0 + 10 \cdot \alpha_1 \log_{10}(d) && d \le d_{BP} \\
L(d) &= L_0 + 10 \cdot \alpha_1 \log_{10}(d_{BP}) + \cdots \\
&\quad 10 \cdot \alpha_2 \log_{10}\left(\frac{d}{d_{BP}}\right) && d > d_{BP}
\end{aligned}
\tag{3}
$$

where L_0 defines the path loss at 1 m in dB, d is the distance in meters, d_{BP} specifies the breakpoint distance in meters, α_1 is the power-distance gradient before the breakpoint, and α_2 is the power-distance gradient after the breakpoint.

The shadow fading component for the path loss model is defined by Equation 4:

$$f(d) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{d^2}{2\sigma^2}} \tag{4}$$

where f(d) represents a Gaussian random variable with zero mean and standard deviation σ. The path loss model parameters for all the channel environments defined by the IEEE 802.11 standardization committee are summarized in the Table 1:

Table 1: Path loss model parameters for IEEE 802.11 models

Model	d_{BP} (m)	α_1	α_2	Standard deviation for shadow fading before d_{BP}	Standard deviation for shadow fading after d_{BP}
A	5	2	3.5	3	4
B	5	2	3.5	3	4
C	5	2	3.5	3	5
D	10	2	3.5	3	5
E	20	2	3.5	3	6
F	30	2	3.5	3	6

In addition to the path loss model parameters, the IEEE 802.11 standardized committee has also provided the PDP for each environment. The PDP gives the information about the fast fading component of the path loss caused due to multipath effects in indoor wireless environment. In this research, the IEEE 802.11 channel model C was used since it best fits our scenario. More details on the choice of this model are provided in section 2.3.1. The PDP for IEEE 802.11 channel model C is a two-cluster model and is depicted by the stem plot shown in Figure 4.

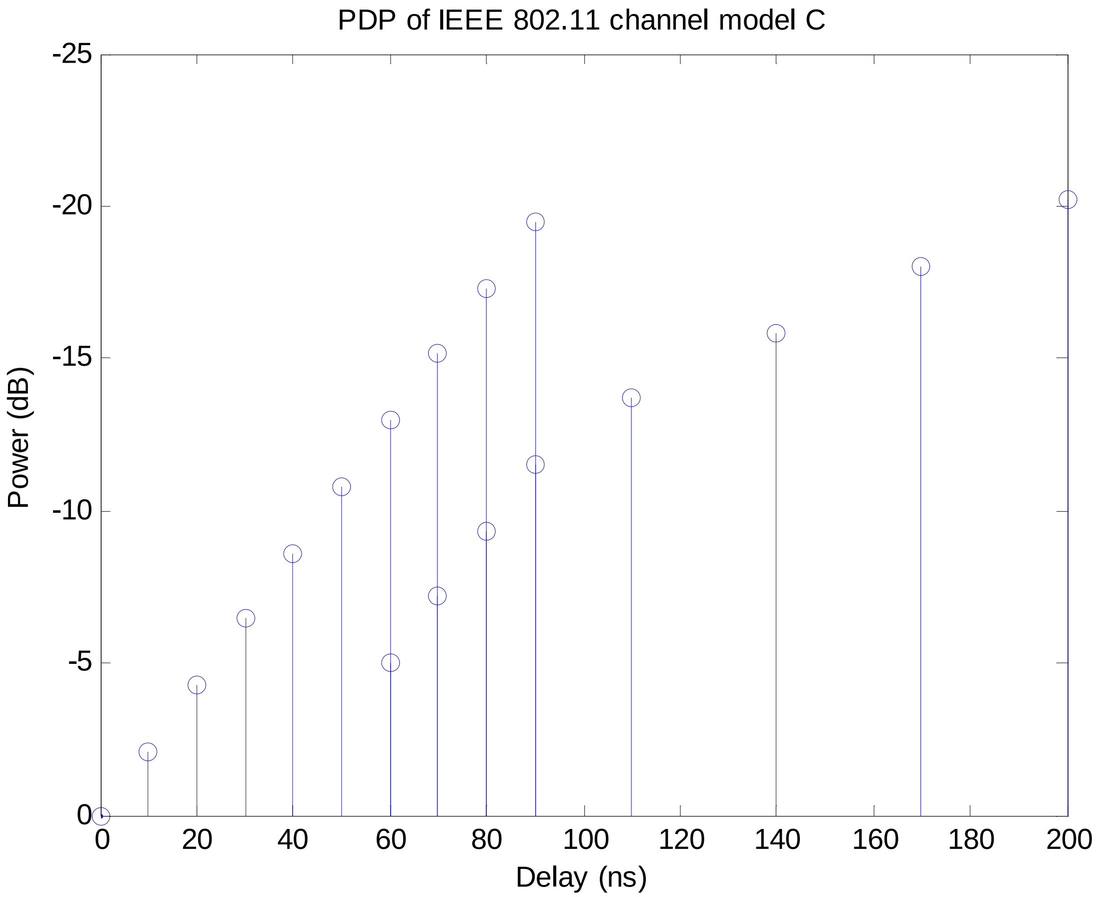


Figure 4: Power Delay Profile of IEEE 802.11 channel model C

The RSS for the IEEE 802.11 channel model can be calculated using Equation 5:

$$RSS\left(d\right)=P_t-L\left(d\right) \tag{5}$$

where P$_t$ specifies the transmitted power in dBm, d is the distance between the access point and the mobile host in meters, L(d) represents the path loss in dB.

2.3.1 Choosing IEEE 802.11 Channel Model for a Wireless Environment

In order to choose the appropriate channel model for a particular wireless environment, the RMS delay spread of that environment needs to be evaluated. The RMS delay spread can either be evaluated from real-time measurements or by the information generated by the RT software. In this research, the RT software was used to evaluate the RMS delay spread of the 3rd floor of the Atwater Kent (AK) building at Worcester Polytechnic Institute. For AK 3rd floor scenario, we generated channel impulse responses using the RT software for different sets of transmitter and receiver locations. These locations were chosen such that the distance between the transmitter and the receiver varies from a small value to a large value. The RMS delay spread was then calculated using Equation 6.

$$\tau_{rms}=\sqrt{\overline{\tau^2}-\left(\overline{\tau}\right)^2}$$
$$\overline{\tau^n}=\frac{\sum_{i=1}^{L}\tau_i^n\left|\beta_i\right|^2}{\sum_{i=1}^{L}\left|\beta_i\right|^2} \tag{6}$$

where β_i represents the amplitude of the ith path, τ_i represents the time taken by the ith path to reach the receiver, and L represent the number of ray paths between the transmitter and the receiver. The RMS delay spread was found to vary between 24 ns and 31 ns. In the IEEE 802.11 standard, channel model C with an RMS delay spread of 30 ns closely represents the AK 3rd floor scenario. Hence, channel model C was used in this research.

2.4 Correlation Model for Shadow Fading

In indoor channel models, the effect of shadow fading also needs to be considered. Shadow fading is caused by obstacles blocking the direct path, reflected path or the refracted path between the transmitter and the receiver. The IEEE 802.11 standardization committee has also provided the standard deviations of log-normal (Gaussian in dB) shadow fading for each channel model. These values have been included in Table 1. The zero-mean Gaussian probability distribution is given by Equation 4.

The shadow fading component can be well approximated by a log-normal distribution. This approximation provides randomly generated values that are added to the path loss model defined in Equation 3. However, in order to apply this approximation to include meaningful information about the channel environment, the correlation properties of the shadow fading process need to be applied. The first correlation model for shadow fading was introduced in [17] and a modified version of this model was later presented in [18]. The correlated log-normal shadow fading can be calculated using Equation 7:

$$f(d) = \rho \cdot f(d-1) + \sqrt{\left(1-\rho^{2}\right)} \cdot N(0,\sigma) \tag{7}$$

where N(0,σ) represents a normal random variable with zero mean and standard deviation σ, and ρ is a correlation coefficient calculated using an exponential autocorrelation function defined by Equation 8. For IEEE 802.11 channel model C, σ = 3 for distances less than the 5 m and σ = 5 for distances greater than 5 m.

$$\rho(\gamma) = E\left[f(d)f(d+\gamma)\right] = \sigma^2 \cdot e^{-\frac{\gamma}{\gamma_0}} \tag{8}$$

where γ is the distance separating two samples, γ_0 is a parameter that can be used to specify correlation at a particular distance.

2.5 Site-Specific IEEE 802.11 Channel Model

The IEEE 802.11 channel model described in section 2.3 and the correlation model for shadow fading presented in section 2.4 were combined to generate a site-specific IEEE 802.11 channel model. The site-specific IEEE 802.11 channel model is represented by Equation 9.

$$\begin{aligned} RSS(d) &= P_t - L_0 - 10 \cdot \alpha_1 \log_{10}(d) + \cdots & d \le d_{BP} \\ &\quad \rho \cdot f(d-1) + \sqrt{1-\rho^2}\, N(0,\sigma_1) \\ RSS(d) &= P_t - L_0 - 10 \cdot \alpha_1 \log_{10}(d_{BP}) - \cdots & d > d_{BP} \\ &\quad 10 \cdot \alpha_2 \log_{10}\left(\frac{d}{d_{BP}}\right) + \rho \cdot f(d-1) + \sqrt{1-\rho^2}\, N(0,\sigma_2) \end{aligned} \tag{9}$$

where,

L_0 defines the path at one meter distance in dB: (L_0 = 31 m),

d is the distance in meters,

d_{BP} specifies the breakpoint distance in meters: (d_{BP} = 5 m),

α_1 is the power-distance gradient before the breakpoint: ($\alpha_1 = 3$),

α_2 is the power-distance gradient after the breakpoint: ($\alpha_2 = 5$),

ρ is the correlation coefficient: ($\rho = 0.96$),

σ1 is the standard deviation of shadow fading process before the breakpoint distance: (σ1 = 3), and

σ2 is the standard deviation of shadow fading process after the breakpoint distance: (σ2 = 5).

The testbed hardware (Propsim) that simulates the channel between the transmitter and the receiver requires both the RSS information and the delay profile. The RSS is calculated using Equation 9, while the path delay profile illustrated in Figure 4 is used. In order to evaluate the accuracy of the site-specific IEEE 802.11 channel model, the RSS data generated by the site-specific IEEE 802.11 channel model was compared with the RSS data generated by the RT channel model.

2.5.1 Calibrating L_0 for IEEE 802.11 Channel Model

Before we can compare the site-specific IEEE 802.11 channel model and the RT channel model, the path loss at one meter for the IEEE 802.11 channel model needs to be calibrated. The path loss at one meter distance (L_0) was calibrated by plotting a two-piece line through the RSS data generated using RT software. The RSS data for RT software was generated in the AK 3rd floor scenario at different distances from the transmitter. Figure 5 represents the configuration of receiver locations in the AK 3rd floor scenario.

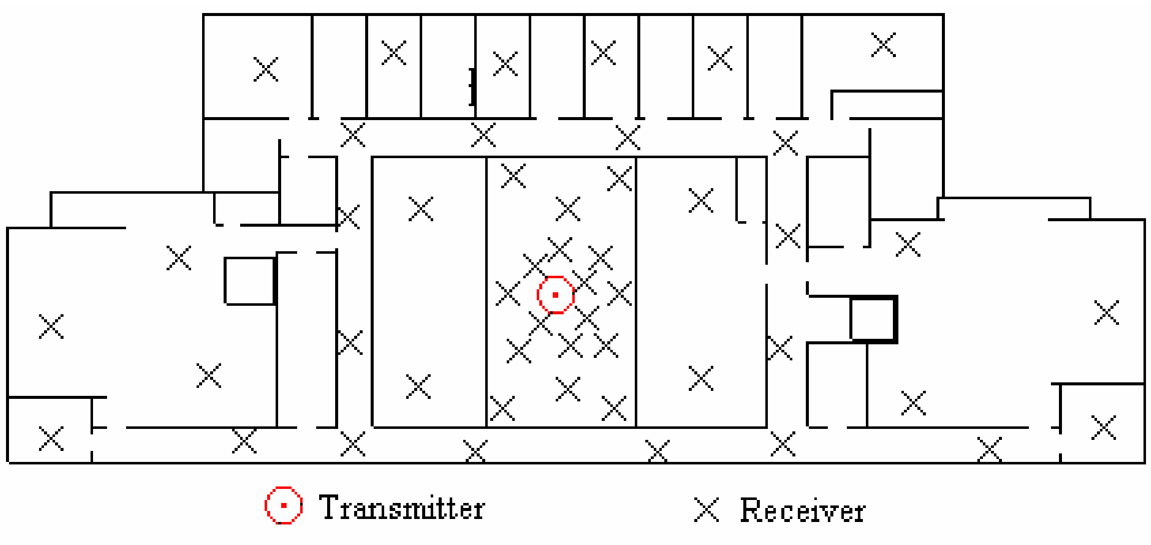


Figure 5: Receiver locations in AK 3rd floor scenario for L_0 calibration

The results of the calibration are shown in Figure 6. The L_0 for RT generated RSS data was found to be around -31 dBm. Hence, the L_0 used for the site-specific IEEE 802.11 channel model is also -31 dBm. It can also be deduced from the results that the signal strength in RT changes at a sharper rate as compared to the IEEE 802.11 channel model for distances greater than 5 meters.

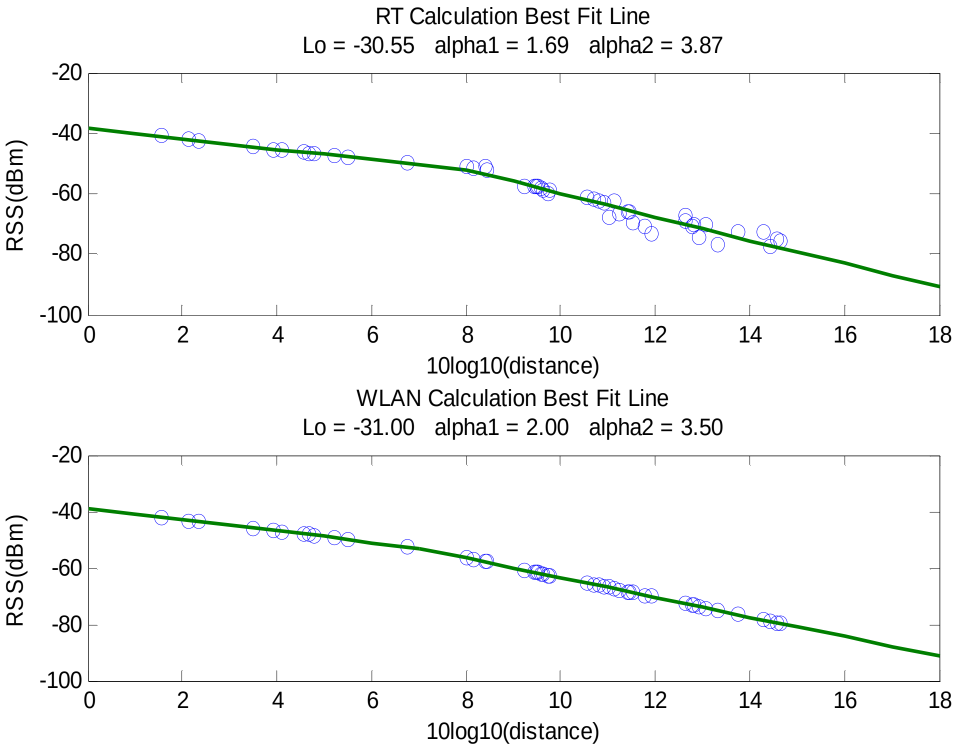


Figure 6: RSS plots for Ray Tracing and IEEE 802.11 channel models

The MATLAB code used to plot the two-piece model is provided in Appendix B.5.

2.5.2 Choosing the Correlation Coefficient for Shadow Fading

In order to choose the correlation coefficient, the RSS data for the site-specific IEEE 802.11 channel model and the RT channel model was generated for the scenario shown in Figure 7. The RSS data for both channel models was then compared side-by-side at different sample points in the scenario as shown in Figure 8. The same process is repeated for different values of the correlation coefficient (ρ) in Equation 9. The value of ρ was then chosen based on how closely the RSS data for the two channel models correlated with each other. At the end of the simulations, the value of ρ was determined to be in the range of 0.92 to 0.98. In this thesis, the correlation coefficient of 0.96 was used for the site-specific IEEE 802.11 channel model.

2.5.3 RSS Comparison of Channel Models

Figure 7 shows the scenario in which the RSS data for the site-specific IEEE 802.11 channel model and the RT channel model was compared.

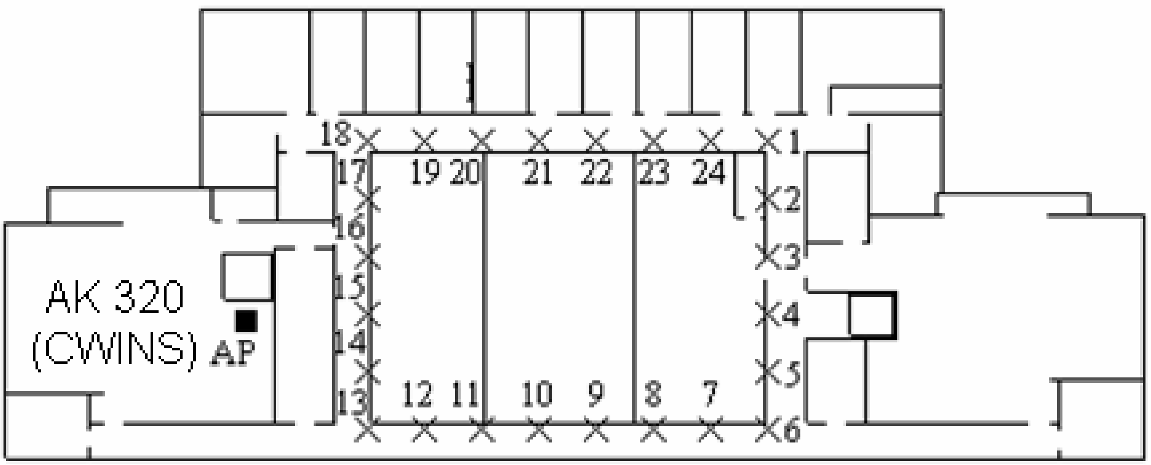


Figure 7: Simulation scenario for RSS comparison of channel models

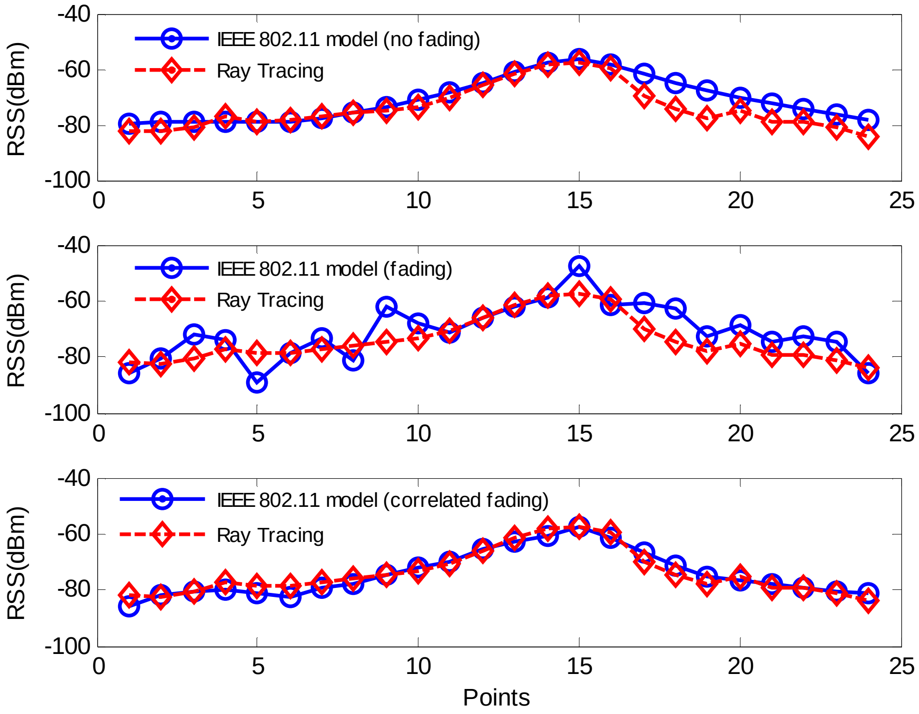


Figure 8: RSS comparison of the site-specific IEEE 802.11 and RT channel models

Figure 8 shows the correlation of the RSS data for the RT channel model and the IEEE 802.11 channel model without shadow fading, with shadow fading and with correlated shadow fading. The MATLAB code used for this plot is included in Appendix B.4.

2.6 Comparison of Ray Tracing and IEEE 802.11 Channel Models

In this section, a comparison of the statistical time-domain IEEE 802.11 channel model and the deterministic building-specific RT channel model is provided. Most of the information in this section has been adapted from [10]. At the end of this section, a table is provided to summarize the pros and cons of the two channel models.

The development of a statistical model for different environments can involve a substantial effort to collect accurate databases of measurements. However, implementation of statistical models on a computer requires minimal computational resources. On the other hand, in general, the development of the deterministic models does not require any measurements to be made. Though, in practice, some measurements are taken to check the accuracy of the model and to determine the model parameters such as conductivity and permittivity of structural materials. The implementation of RT channel model always requires extensive computational resources.

Computation time required for processing the RT channel model grows exponentially with the complexity of the details included in the layout of the area. The computation time required does not necessarily increase with the size of the area; hence the computation time required for modeling a small area with many walls might be as great as for a large area containing only a few walls. The application of the RT technique will in general involve different analytical approaches for different application areas. Statistical models have the advantage of being computationally efficient and easy to generalize. Furthermore, the development of a statistical model is independent of the layout and structural details of the coverage area which removes the requirement of the building layouts for each application. A disadvantage of the statistical models is that they cannot provide the relationship between the layout of a building and the detailed channel response in a specific location. “Applications such as the optimum siting of antennas, or the analysis of systems using sectored antennas, are best served by use of deterministic

RT models, which can provide realistic estimates of the azimuthal distribution of rays received in a multipath environment [10].”

The actual implementation procedure of the two channel models in the testbed are discussed in more detail in Chapter 4. However, some of the experiences are discussed here. For IEEE 802.11 channel model, the received power at different receiver locations from the transmitter are easily generated using equation 4. The PDP used for all these locations is also the same, hence minimum computation is required. On the other hand, RT software provides a separate PDP for each receiver location. In addition, the PDP results need to be modified before they can be used in the testbed. Also, the received power information also needs to be computed from the results provided by the RT software for every receiver location; hence in comparison to IEEE 802.11 model more computations are required. In section 2.5, spatial correlated shadow fading was implemented for the IEEE 802.11 model and it was concluded that the RSS results provided by the updated channel model closely fits the RSS results obtained from RT channel model. These results are illustrated in Figure 8.

Table 2 summarizes the pros and cons of the two channel models discussed in this chapter.

Table 2: Pros and cons of the IEEE 802.11 and RT channel models [10, 19]

METHOD	PROS	CONS
Building-Specific Deterministic Ray Tracing Channel Model	• Development of deterministic model does not require any real-time measurements • No real-time measurements are needed for implementation • Higher resolution is achievable by using a finer grid with no substantial cost increase • Changes in network geometry are reflected in a new set of simulations • A customized resolution for different applications can be provided	• Requires extensive computational resources • Need a third party software to generate the model • An accurate floor plan of the building is needed • The location of all AP’s must be known in advance • Generally less accurate compared to the real-time on site measurements
Statistical Time-Domain IEEE 802.11 Channel Model	• Implementation requires minimal computational resources • Development of this model is independent of the building layout or an accurate floor plan • No real-time measurements are needed for implementation • Higher resolution is achievable by using a finer grid with no substantial cost increase • Changes in network geometry are reflected in a new set of simulations • A customized resolution for different applications can be provided	• Development of a statistical model requires substantial effort • The location of all AP’s must be known in advance • Generally less accurate compared to ray tracing and the real-time on site measurements

Chapter 3: RFID Localization in RTLS

In this chapter, an outline of the RFID localization in RTLS systems is provided. Section 3.1 provides an overview of traditional localization techniques presented in the literature and used by the RFID manufacturers. Section 3.2 provides an overview of traditional RFID devices, summary of RTLS standards, and a brief overview of emerging RFID technologies.

3.1 Overview of Localization Techniques

Localization, also referred to as positioning, was first implemented when a series of satellites were launched for the Global Positioning System (GPS). Although, GPS is widely used for personal and commercial outdoor applications, it does not perform satisfactorily in indoor areas [20]. Predicting the location of an individual or an object in an indoor environment can be a difficult task, often producing ambiguous results, due to the harsh wireless propagation environment in most such areas. The indoor radio propagation is characterized as site-specific, exhibiting severe multipath and low probability of Line of Sight (LOS) signal propagation between the transmitter and receiver, making accurate indoor positioning very challenging [10].

In general, localization techniques utilize metrics of the received radio signals. Some traditional localization techniques include Angle of Arrival (AOA), RSS, Time of Arrival (TOA), and Time Difference of Arrival (TDOA). Respective channel models targeting the behavior of each of these metrics have been developed and are available in the literature. However, models for indoor localization applications must account for the effects of the harsh indoor wireless channel behavior on the characteristics of these metrics. In this section we provide a brief overview of the different localization techniques [21].

There are three main entities found in any positioning system, namely, location sensing devices, positioning algorithm and a display system. The location sensing devices measure the distance-related metrics between themselves and a fixed Reference Point. As discussed above, these metrics can be based on TOA, AOA, RSS, and TDOA techniques. The positioning algorithm processes the reported metrics to estimate the location coordinates of the location sensing devices (such as laptops, PDAs and RFID tags). The display system exhibits the location of the mobile terminal to the user. The accuracy of the location estimate depends on the accuracy of the localization metrics used and the complexity of the positioning algorithm. Figure 9 shows a basic block diagram of a positioning system [21].

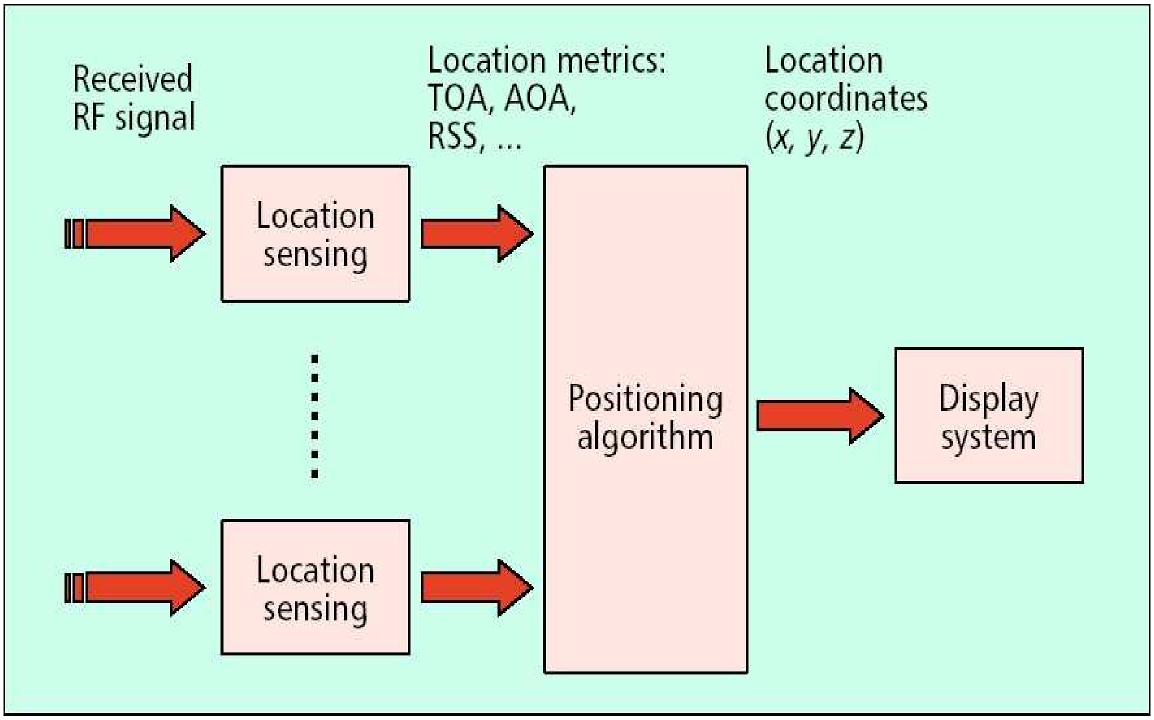


Figure 9: Block diagram of a positioning system

3.1.1 Time of Arrival (TOA)

In TOA-based localization, time of arrival of the First Detected Peak (FDP) of the received signal is used to determine the time of flight between the transmitter and the receiver and consequently the distance between the transmitter and the receiver [22]. Channel Impulse Response (CIR) is generated by RT software for different locations of transmitter and receiver. The ideal channel impulse response is usually referred to as the infinite-bandwidth channel profile since the receiver could theoretically acquire every detectable path. In practice, however, the channel bandwidth is limited. Hence, the CIR is filtered with a limited bandwidth filter to obtain paths with pulse shapes. It can be shown that the sufficient bandwidth for accurate RFID localization based on the TOA metric is around 200 MHz [23]. Figure 10 shows how a channel impulse response for indoor environment is used for TOA localization.

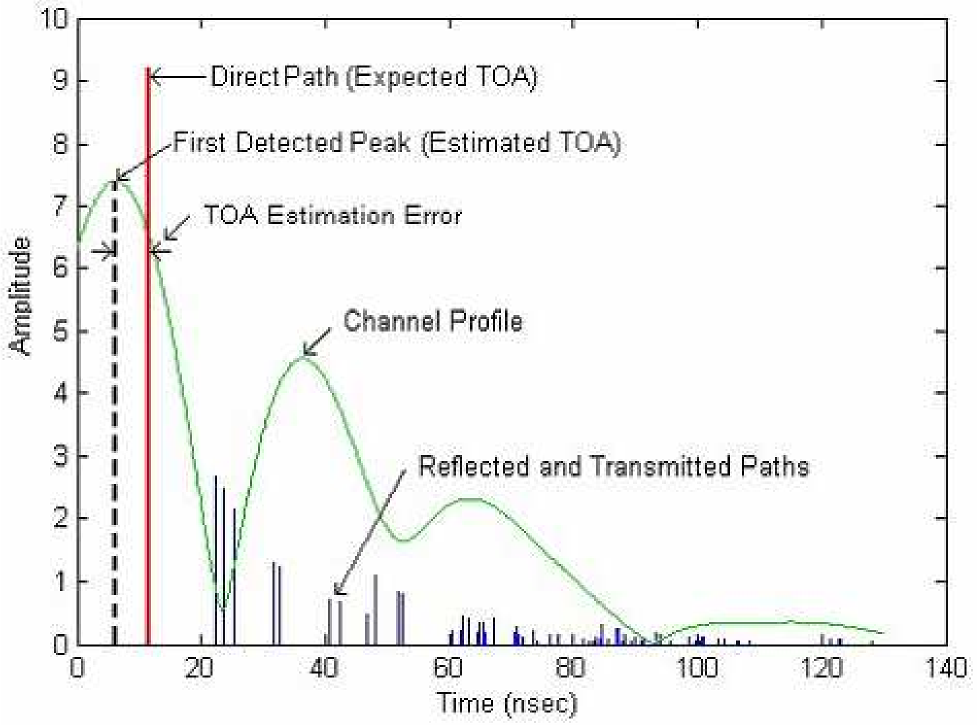


Figure 10: Overview of channel impulse response for TOA localization [24]

The TOA localization suffers from two sources of errors: multipath effects and Undetected Direct Path (UDP) conditions. Multipath effects result in the reflected and transmitted paths being received along with the direct path, hence causing ranging errors. This error can be reduced by increasing the transmission bandwidth of the system; as bandwidth increases, the pulses become narrower and the TOA estimate narrows in on the expected TOA of the direct path. The UDP condition occurs when the direct path falls below the detection threshold of the receiver. This condition usually happens at the edges of the coverage area or when the direct path between the transmitter and the receiver is blocked by large metallic objects. The UDP condition results in very large ranging errors and cannot be minimized by increasing the bandwidth of the transmission system [22].

3.1.2 Received Signal Strength (RSS)

RSS localization technique is the simplest approach for locating a mobile host (MH) in WLAN environment. In this technique, the received powers from the wireless reference points, such as access points, are used to collect data and then use this information to determine the location of an object. This major advantage of this technique is its simplicity. RSS is a simple metric that can be measured easily and it is measured and reported by most wireless devices. For example, the MAC layer of IEEE 802.11 WLAN standard provides RSS information from all active access points (APs) in a quasi-periodic beacon signal that can be used as a metric for localization [22]. Figure 11 provides an overview of the RSS localization technique.

The RSS localization technique suffers from larger ranging errors as compared to TOA localization technique, but these errors can be reduced by increasing the density of access points in a particular indoor environment. This is the leading localization technique used by RFID manufacturers.

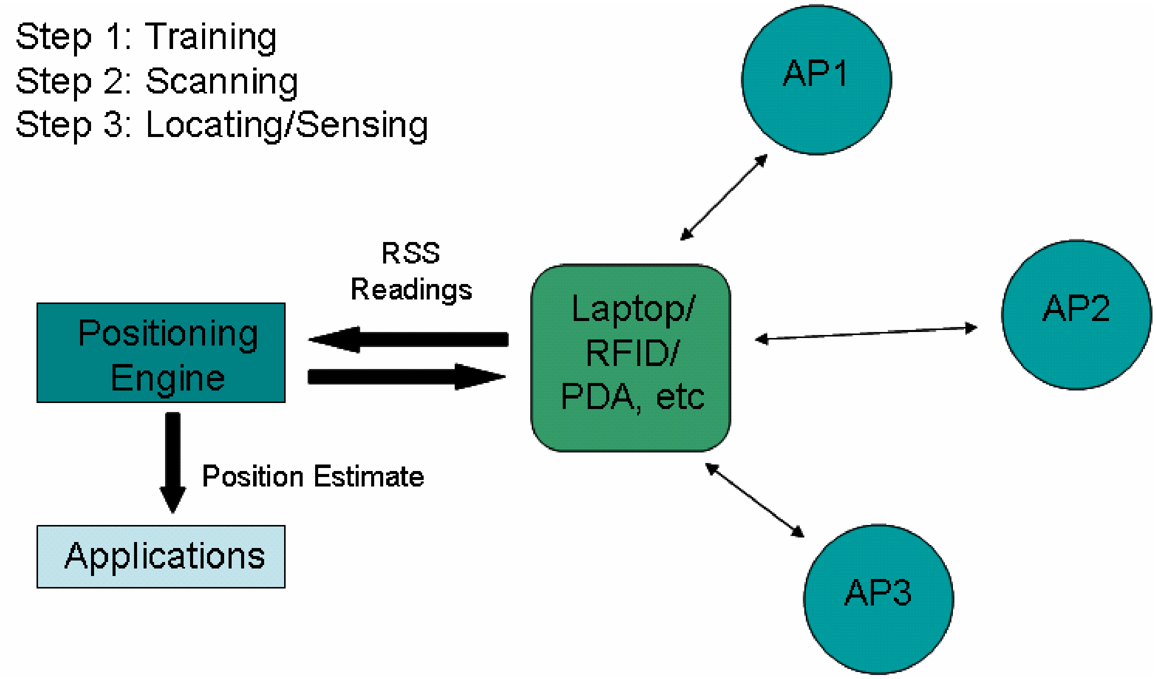


Figure 11: RSS Localization Technique

3.1.3 Angle of Arrival (AOA)

As the name implies, AOA gives an indication of the direction of the received signal. In order to estimate the AOA, the RFID tag and/or RPs must be equipped with a special antenna structure, specifically, an antenna array. In practice, severe multipath conditions in an indoor environment can lead to erroneous estimation of AOA. Thus, relying on the AOA techniques alone in those environments would result in location estimates of low accuracy [22]. However, AOA estimation can be employed in adverse channel conditions to enhance the estimation of TOA.

3.1.4 Time Difference of Arrival (TDOA)

The TDOA technique works by measuring the exact time of arrival of a transmitted signal at three or more separate receiver locations. Because radio waves travel at a fixed known rate (the speed of light), by calculating the difference in the arrival times at pairs of receiver locations, it is possible to calculate the hyperbolas in which the transmitting device is located. This technique can work with any transmitting device and only requires modifications to the receiver locations [25].

3.2 RFID Technologies

Radio Frequency Identification (RFID) is an automatic identification technology that relies on storing and remotely retrieving data using devices called RFID tags. The RFID tags are essentially radio transponders. Recently, there has been an enormous amount of innovative development of RFID devices, and the emerging RFID devices are being integrated with traditional localization techniques, especially in RTLS [10]. Section 3.2.1 provides an overview of the traditional RFID technologies. Section 3.2.2 describes the use of RFID technologies in RTLS and the notion of Internet of Things associated with these technologies. Section 3.2.3 includes some of the emerging RFID technologies that can be used for indoor localization and can be an integral part of an RTLS.

3.2.1 Traditional RFID Tags

The traditional RFID tags are generally used for storing and retrieving data for identification and tracking purposes. These tags can be attached to or incorporated in a product, animal, or person for the purpose of identification using radio waves.

Traditional RFID come in three general varieties: passive, semi-passive (also known as semi-active), or active [26].

3.2.1.1 Passive RFID

Passive RFID tags have no internal power supply. The CMOS integrated circuit in the tag gets enough power from the minute electrical current induced in the antenna by the incoming radio signal to power up and transmit a response. Most passive tags signal by backscattering the carrier signal from the reader. This means that the antenna has to be designed collect not only the power from the incoming signal but to also transmit the outbound backscatter signal. The passive RFID tag can also contain a non-volatile EEPROM that can be used for storing data, hence it is not limited to just store an ID number. The lack of onboard power supply means that the device can be quite small. The products available commercially can be embedded in a sticker or under the skin in case of low frequency RFID tags. Some of the manufacturers who are developing passive RFID tags include Hitachi Ltd., SmartCode, Alien Technology and Symbol Technologies.

3.2.1.2 Active RFID

Active RFID tags have their own internal power source which is used to power any integrated circuits that generate the outgoing signal. Active tags are capable of establishing a connection with a reader independently and are hence more reliable than the passive tags. Active tags also transmit at higher power levels than passive tags, allowing them to more effective in “RF challenged” environments like water (including humans/cattle, which are mostly water), metal (shipping containers, vehicles), or at longer distances. Active tags typically have much larger range (up to 100 m) and larger memories than passive tags, as well as the ability to store additional information sent by the transceiver. Some active RFID tags include sensors such as temperature logging, which have been used in concrete maturity monitoring or to monitor temperature of perishable goods. Other such sensors associated with active RFID tags include humidity, shock/vibration, light, radiation, temperature, and atmospherics like ethylene. At present, the smallest active tags are about the size of a cold capsule.

The active RFID tags used in this research are used to convey the received power from the access points (wireless reference points) to positioning software installed in a laptop. These tags are based on different wireless standards, such as IEEE 802.11 wireless LAN standard, are incorporated with traditional localization techniques, such as RSS and TOA techniques, and are an integral part of RTLS.

3.2.2 Real-time Locating Systems (RTLS)

RTLS are electronic systems that are intended to locate small electronic devices, people, or things at any time. In order to be called an RTLS, the system must be able to provide real time identity and location of people and assets when required. RTLS systems can track assets without being in line-of-sight inside a building. They can provide real-time information to authorized users via a corporate intranet or through network software. They can solve expensive logistics problems by instantly locating assets or persons. They can help in maintaining a complete log of movements for auditing, security and usage analysis. They can generate real-time inventory of all tagged assets, thus saving time and money in locating an asset [27].

RTLS systems usually comprise of active RFID tags and positioning software that can be used to locate and track assets or personnel. RTLS can be used by companies in the healthcare, oil and gas, chemical, government and automotive industries to save considerable amount of money by providing improved asset management and tracking. Recently, there have been a myriad of emerging RTLS solutions based on different positioning technologies with diverse specifications. In the next section, a brief overview of the RTLS standard defined by the International Standards Organization (ISO) is provided.

3.2.2.1 ISO/IEC 24730 RTLS standard

ISO/IEC 24730 defines two air interface protocols and a single application program interface (API) for RTLS for use in asset management and is intended to allow for compatibility and to encourage interoperability of products for the growing RTLS market. In order to be compliant with this standard, RTLS must comply with ISO/IEC 24730:1-2006, the RTLS API, and at least one air interface protocol defined in ISO/IEC 24730:2-2006 [2].

The ISO/IEC 24730 standard is divided into two parts. The first is the API, which is general enough for any number of different RTLS products to use it to share location data and information with each other. The API uses widely known Web-based standards including eXtensible Markup Language (XML) and SOAP, a protocol for exchanging

XML-based messages over computer networks. The second part of the standard describes the air-interface protocol in three sections. The first section addresses the tag-reader air interface at 2.4 GHz and defines a networked location system providing x-y coordinates. The second section describes the air interface for low frequency (around 120 kHz) devices, known as exciters, which provide a location stamp when an RTLS system is not using multiple 2.4 GHz interrogators to determine location. The third interface is the On/Off Key (OOK) which allows a tag to respond simply to a handheld device. The OOK enables an active 2.4 GHz RTLS tag to communicate with a simple, less expensive handheld reader [28].

3.2.3 Emerging RFID Technologies

Recently, there has been a multitude of emerging RFID technologies. These emerging RFID technologies are based on a number of wireless interoperability standards, which specify varied requirements as to power level, modulation technique, occupied bandwidth, frequency of operation, and so forth. They also utilize a variety of PHY and MAC layer techniques. Examples of these technologies include WiFi RFID, UWB RFID, and ZigBee RFID. The following sections provide a brief overview of these technologies [1].

3.2.3.1 WiFi RFID

The WiFi RFID tags are based on the IEEE 802.11 a,b,g,n wireless standard. These tags operate in the 2.4 GHz frequency band and have a power requirement of about 100 mW. Most of these RFID tags use RSS localization technique. Hence, the bandwidth of these tags is around 20 MHz. Some of the manufacturers who are developing this RFID technology are Ekahau, PanGo Networks, G2 Microsystems, AeroScount and WhereNet.

3.2.3.2 UWB RFID

The UWB RFID tags are based on the IEEE 802.15.3 wireless standard. These tags operate in the 3.4 – 10.6 GHz frequency band and have a power requirement of about 200 mW. Most of these RFID tags use TOA localization technique. Hence, they have large bandwidths which are in the range of 500 – 3000 MHz. Mutlispectrial Solutions Inc. (MSSI) is the only manufacturer who is currently developing this TFID technology.

3.2.3.3 ZigBee RFID

The ZigBee RFID tags are based on the IEEE 802.15.4 wireless standard. These tags operate at the frequencies of 898 MHz, 915 MHz, and 2415 MHz. These tags have low bandwidths which are in the range of 2 – 5 MHz. The most important feature of these tags is their low power consumption of about 0.5 mW which gives these tags a long battery life. This RFID technology is currently being developed by Bulldog Technologies.

Chapter 4: Design and Implementation of the Testbed

In this chapter, we discuss the design and implementation of the laboratory testbed used for performance evaluation of RFID technologies in a controlled and repeatable environment. In section 4.1, the overall architecture of the testbed has been presented. Section 4.2 provides an overview of the different elements found in the testbed. Section 4.3 presents the test procedures for using the testbed.

4.1 Testbed Architecture

The RT software (PlaceTool) and the MATLAB Instrument Control Toolbox (ICT) are implemented in the Personal Computer (PC). The RT channel model and the site-specific IEEE 802.11 channel model are both generated in the PC and transferred to the Propsim channel emulator through the Local Area Network (LAN) interface. The WiFi access points are connected to the input ports of the Propsim, while its output port is connected to the laptop and the WiFi RFID through a RF circulator. The Propsim simulates the channel between the access point and the WiFi RFID using the channel models supplied by the user. The RFID reads the output power from the Propsim and relays it to the Ekahau software which is installed in the laptop. The output signal of the Propsim contains RSS information from the access points. The Ekahau software uses this information first to calibrate its database and then for tracking and locating mobile clients. A real-time representation of the testbed is shown in Figure 12 and a detailed schematic of the testbed is presented in Figure 13.

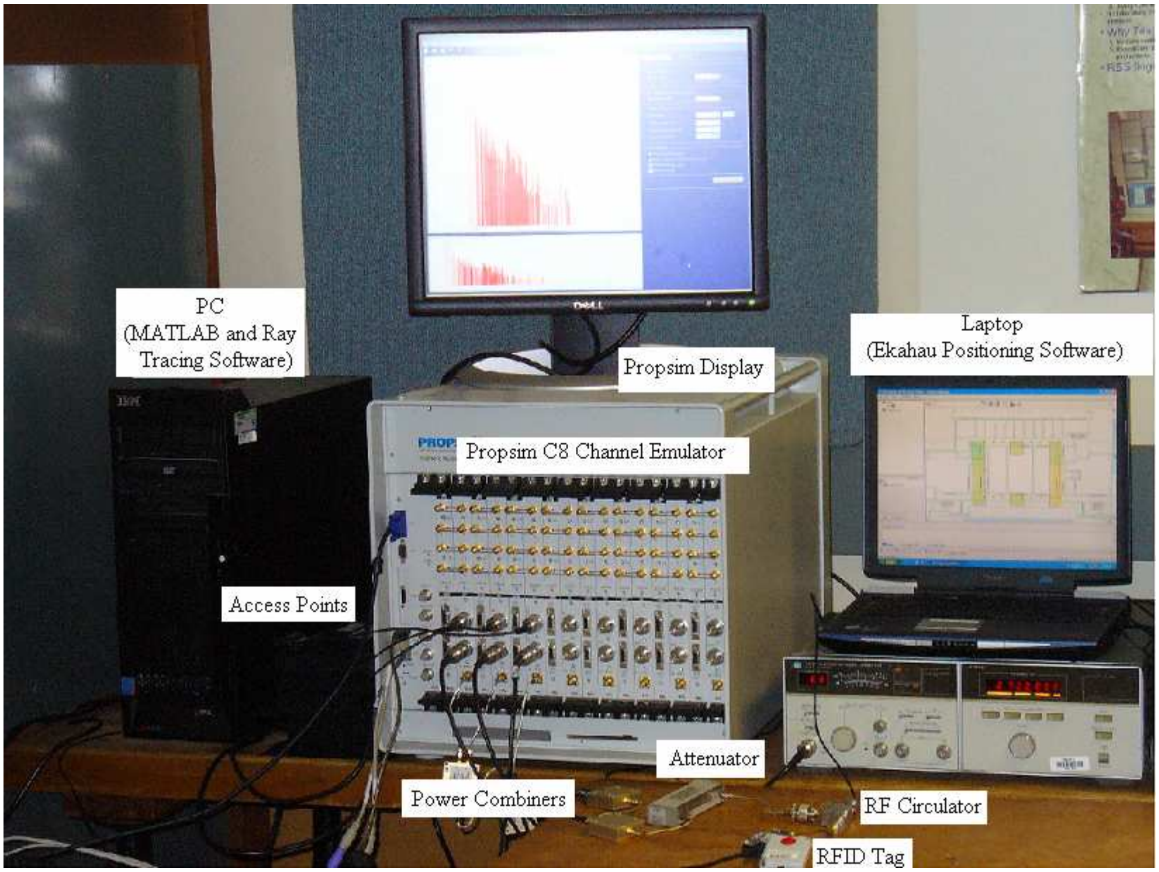

PC
(MATLAB and Ray Tracing Software)
Propsim Display
Laptop
(Ekahau Positioning Software)
Propsim C8 Channel Emulator
Access Points
Attenuator
Power Combiners
RF Circulator
RFID Tag

Figure 12: Actual representation of the testbed

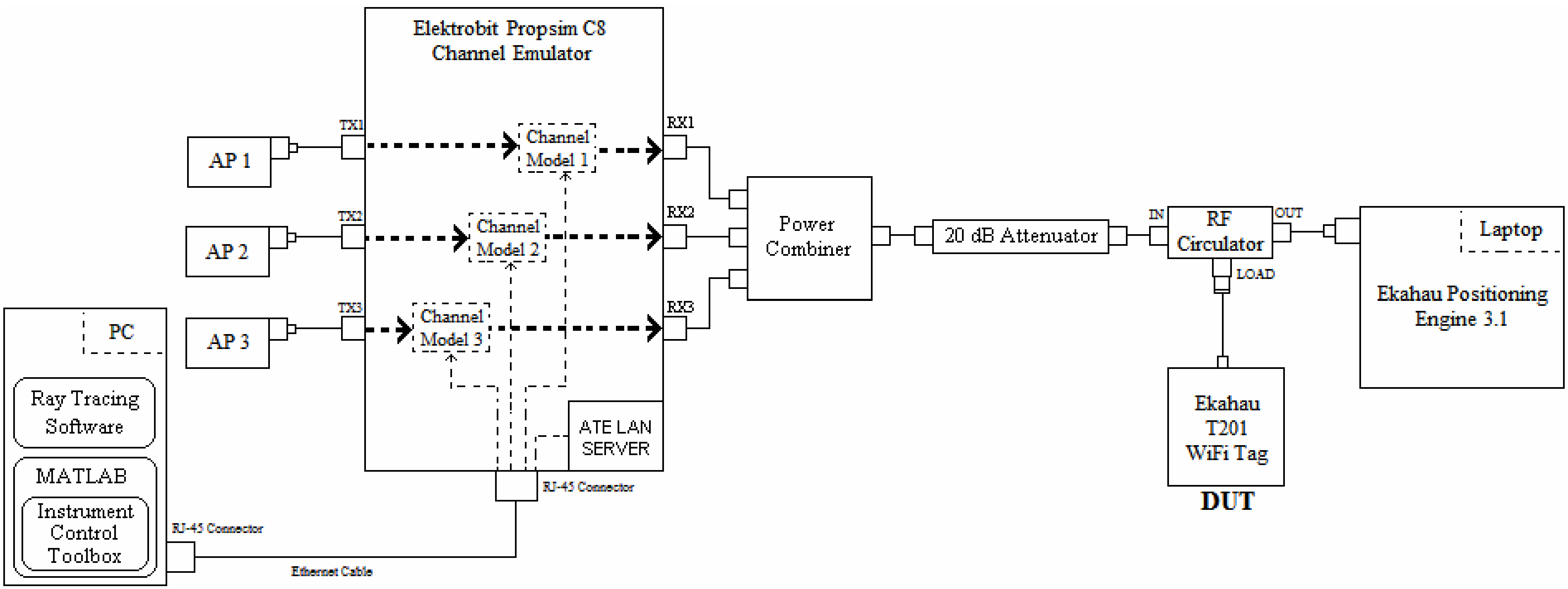


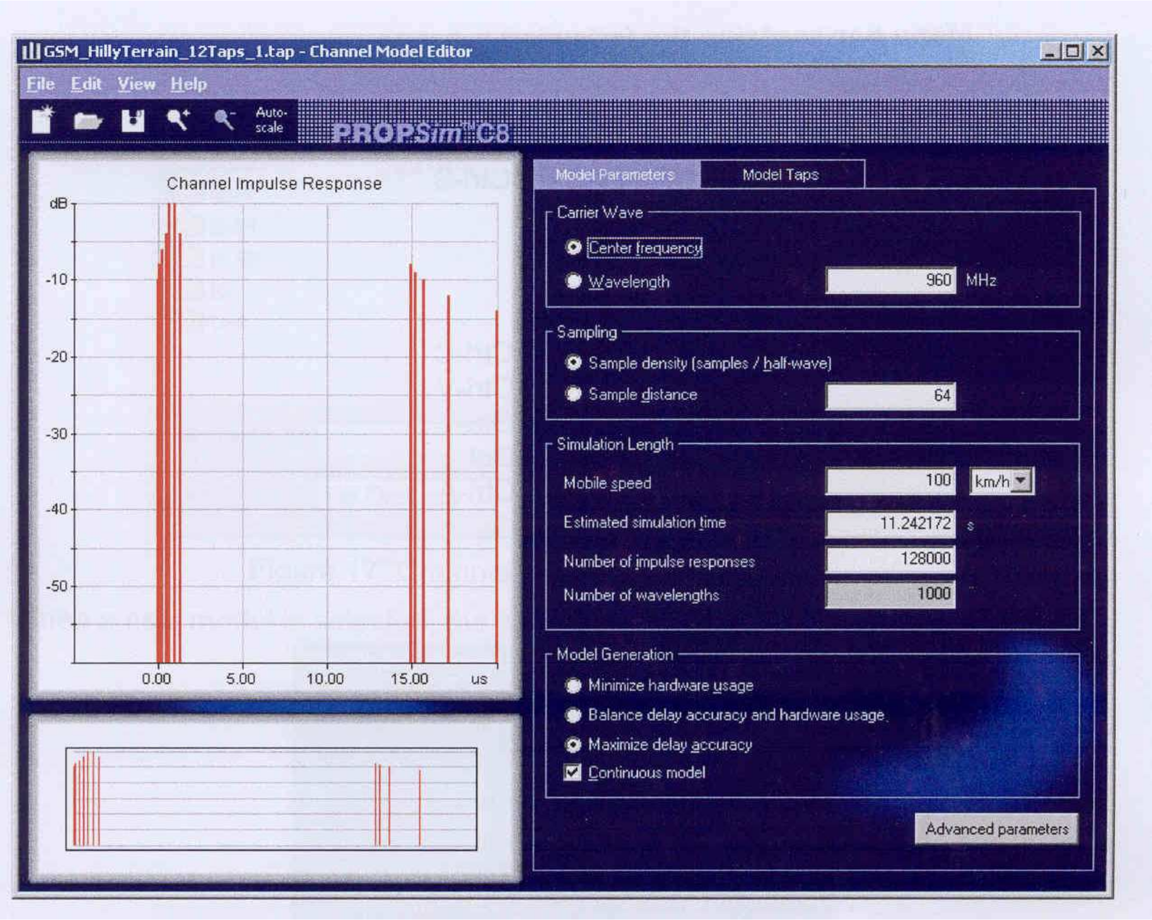
Figure 13: Detailed Schematic of the Testbed

4.2 Elements of the Testbed

The testbed consists of three main elements: Propsim real-time channel emulator, Ekahau positioning software, and Ekahau WiFi RFID tag. In this section, the different elements found in the testbed are described briefly.

4.2.1 Real-time Channel Emulator

Propsim C8 is a real-time hardware radio channel emulator developed by Elektrobit that can emulate up to eight independent or correlative wideband radio channels at a time [29]. It uses a tapped-delay line model to simulate the channel between the input and output of each Propsim channel. Each Propsim channel has a bandwidth of 70 MHz, which is sufficient for modeling RSS based RFID localization technique. Figure 14 shows the Propsim user interface used to define the channel model between the access points and the RFID tag.



Carrier Wave
- ● Center frequency
- ○ Wavelength: 960 MHz

Sampling
- ● Sample density (samples / half-wave)
- ○ Sample distance: 64

Simulation Length
- Mobile speed: 100 km/h
- Estimated simulation time: 11.242172 s
- Number of impulse responses: 128000
- Number of wavelengths: 1000

Model Generation
- ○ Minimize hardware usage
- ○ Balance delay accuracy and hardware usage
- ● Maximize delay accuracy
- ☑ Continuous model

Figure 14: Propsim user interface used to define channel model

4.2.2 RSS Positioning Software

Ekahau Positioning Engine (EPE) is a software based RTLS solution developed by Ekahau [30]. It uses RSS fingerprinting method to locate an object by communicating with the access points and WiFi RFID devices present in the building. EPE consists of a database that is calibrated by storing signal strength measurements at several locations (training points) in the floor plan. The calibration measurements can be based on either access points or RFID devices present in the building. Ekahau Manager provides the user interface for the EPE that first facilitates the calibration process and is later used for localization purposes. EPE and Ekahau Manager are implemented in the laptop.

4.2.3   WiFi RFID Tag

Ekahau T201 WiFi RFID Tag is a small active radio tag that can be used for real-time tracking of people and assets in any standard 802.11b/g network. The tag has a medium access control of Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) and a transmission technology based on Direct Sequence Spread Spectrum (DSSS). It operates in the frequency band of 2.412-2.484 GHz and has a typical operating range of about 100 m in open space and 40 m in a closed office environment [31]. Figure 15 shows a picture of the RFID tag.

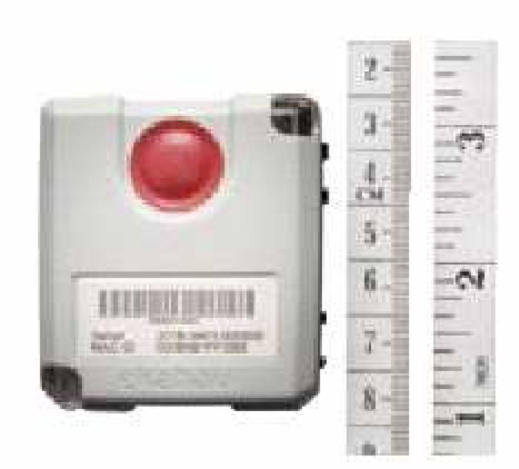


Figure 15: Ekahau T201 WiFi RFID

The WiFi RFID tag has a W.FL antenna connection that can be used to connect to the other elements in the testbed.

4.2.4 Interface between Testbed Components

This section lists the other elements found in the testbed. It also describes in details the connections between each of the testbed elements.

4.2.4.1 Access Points

There are three access points used in this testbed. The access points are connected separately to the first three TX ports on the Propsim channel emulator. Each access point has a PCMCIA card with a port for external antenna connection implemented with an MC connector. The access points are connected to the Propsim channel emulator through an MC-to-N(male) adaptor cable.

4.2.4.2 PC

PlaceTool and MATLAB softwares are installed in the PC. PlaceTool is used to generate the RT channel models. MATLAB is used to generate the site-specific IEEE 802.11 channel models. In this testbed, the site-specific IEEE 802.11 channel models are used for simulating the channel between the access points and the RFID tag. The channel models are supplied to the Propsim channel emulator through the LAN interface. The PC and the Propsim are connected through a RJ-45 Ethernet cable. The MATLAB software also has the license for the Instrument Control Toolbox which can be used to implement an automated connection between the PC and the Propsim.

4.2.4.3 Power Combiner

The three RX ports on the Propsim are connected to a power combiner. The power combiner is used to combine the three channels into a single channel. Since the access points have unique MAC addresses, the signal from each access point can be distinguished. The RX ports on the Propsim are connected to the input ports of the power combiner using N(male)-to-SMA(male) adaptor cables.

4.2.4.4 Variable Attenuator

The variable attenuator is set to 20 dB attenuation. The attenuator is used to reduce the RSS readings from each of the three access points by 20 dB. The external attenuator is required to implement appropriate path loss for each propsim channel since the attenuation in the propsim is limited to -45 dB. The output port of the power combiner is connected to the variable attenuator through a SMA(male)-to-SMA(male) adaptor cable.

4.2.4.5 RF Circulator

The RF circulator has three ports, namely, IN, OUT and LOAD ports. The IN and OUT ports are one-directional ports. The LOAD port is a bi-directional port. The combined attenuated RF signal from the propsim is supplied to the RFID tag and the transmitted signal from the RFID tag is connected to the PCMCIA card on the laptop. The output port of the variable attenuator is connected to the IN port of the RF circulator through a SMA(male)-to-N(male) adaptor cable. The LOAD port of the RF circulator is connected to the RFID tag through a N(male)-to-SMA(male)-to-SMA(female)-to-W.FL adaptor cable.

4.2.4.6 Laptop

The Ekahau Manager and the EPE are implemented in the laptop. The wireless connection in the laptop is provided by a PCMCIA card. The PCMCIA card has a port for external antenna connection which is implemented with an MC connector. The OUT port of the RF circulator is connected to the laptop PCMCIA card through a N(male)-to-MC adaptor cable.

4.3 Test Procedure

In this section, a step-by-step test procedure is presented for evaluating the localization performance of different RFID tags.. The following sections describe the main steps used in the evaluation process:

Step 1: Choose a Test Scenario

Start the performance evaluation test. Choose a test scenario, i.e. the floor plan, configuration of access points and reference points, and the number of training points. Generate channel models for the test scenario in MATLAB. Extract RSS vs. distance data for sample points (including the training points) for all 3 APs using MATLAB.

Step 2: Configure Propsim Channel Emulator

Open propsim simulation file and set up a 3-to-3 SISO model using channel simulator control in propsim. Configure the three channels in the propsim simulation file with the PDP of IEEE 802.11 channel model C. Set the local oscillator (RFLO) appropriately on the propsim for all the three channels. Verify that the different elements of the testbed are properly connected.

Step 3: Configure Ekahau Positioning Software

Open a new positioning model in Ekahau and load the floor plan of the test scenario. Draw path rails and mark training points and sample points on the floor plan map. If you are training with the RFID tag, set the training preferences in the **File -> Preferences** menu item to laptop (localhost). If you are not training with the RFID tag, set the training preferences in the **File -> Preferences** menu item to RFID tag.

Step 4: Train Ekahau Software

Start the training procedure of the Ekahau software. In Propsim, configure output gains of all the 3 channels to correctly represent the RSS value of a certain receiver location. Verify that the RSS values received in the Ekahau Manager correspond to the channel settings in the propsim. Train the appropriate receiver location in the Ekahau Manager. Repeat this process for other training points chosen for this scenario. After the training process is completed, upload the positioning model to the Ekahau Positioning Engine by choosing **File -> Save to the Postioning Engine** menu item in the Ekahau Manager.

Step 5: Generate Performance Evaluation Results

Start the performance evaluation of your chosen wireless device(s). Turn on tracking of the wireless device(s) in Ekahau Manager. Choose a new location for your receiver location. In Propsim, configure output gains of all the 3 channels to correctly represent the RSS value of the receiver location. Verify RSS values in the Ekahau Manager. Take multiple readings of the predicted location coordinates from Ekahau Manager. Calculate the localization error for the receiver location by finding the ranging error between the actual and the estimated location coordinates. Repeat this step to evaluate the localization performance of your wireless device(s) at other receiver locations. Calculate the mean localization error and generate other statistics. You are done with the performance evaluation test. If you want to test another RFID device for the same scenario, go back to step 4 and connect the new RFID device.

Chapter 5: Results of Performance Evaluation

This section presents the results that were obtained in the experiments conducted for the evaluation of localization performance. Section 5.1 presents the results of localization performance for RayTracing and the site-specific IEEE 802.11 channel models. Section 5.2 presents the results of localization performance of two different WiFi RFID devices.

5.1 Localization Performance of Ray Tracing and IEEE 802.11 Channel Models

In this section, the localization performance of the RT and the site-specific IEEE 802.11 channel models are analyzed.

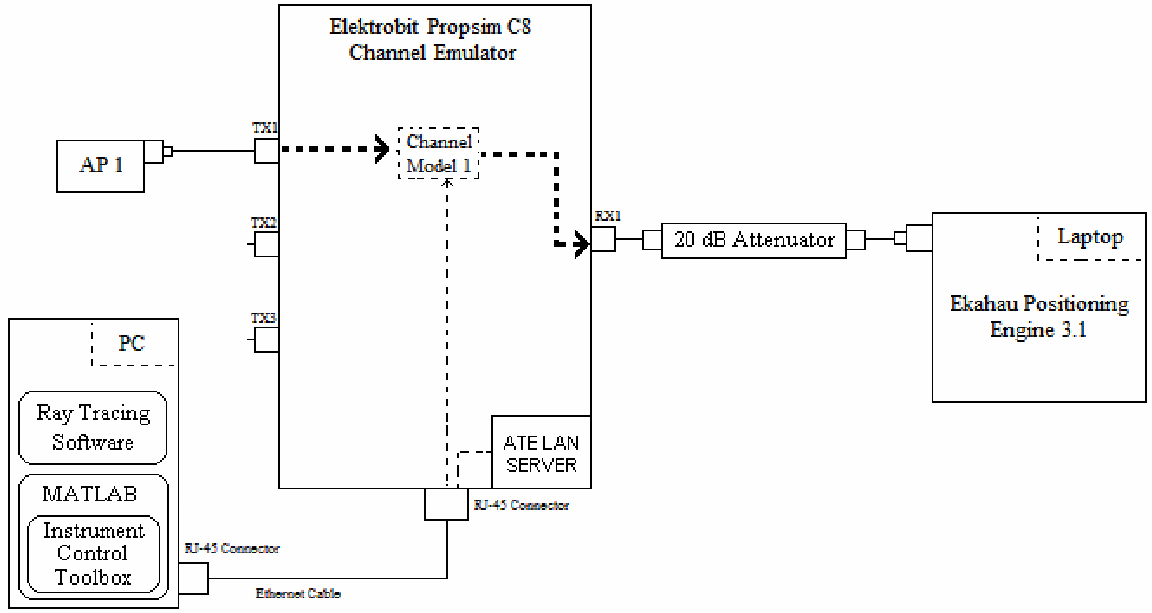


Figure 16: Testbed configuration used for performance evaluation of channel models

For this analysis, the RF circulator and the WiFi RFID tag were removed from the testbed, output of the Propsim was directly connected to the laptop and only one access point was used. The testbed configuration used is shown in Figure 16. The experiment was conducted for the third floor scenario of the Atwater Kent building at Worcester Polytechnic Institute. First, the RT channel model was supplied to the Propsim to simulate the channel between the access point and the laptop. The Ekahau software was trained at eight different locations on the floor plan. The localization performance of the channel model was then evaluated by calculating the mean error between the actual location and the location predicted by the Ekahau software. This experiment was then repeated for the site-specific IEEE 802.11 channel model. Figure 17 shows the location of the access point and the training points used in this experiment.

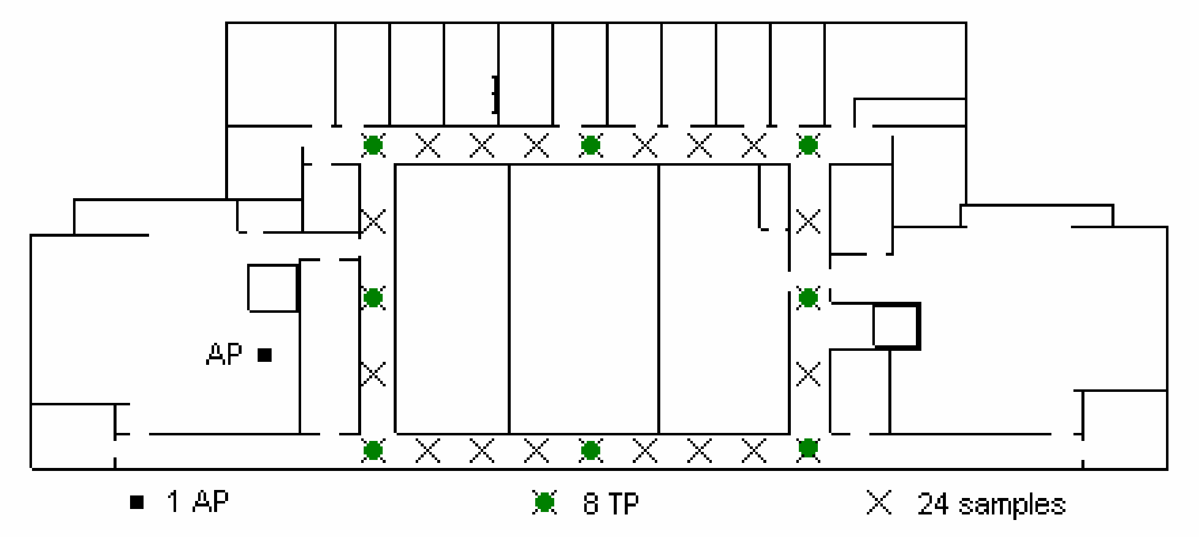


Figure 17: Configuration of access point and training points for comparing localization performance of the channel models

In this experiment, it was found that the mean localization error is 4.8m for IEEE 802.11 channel model and 5.9m for RT channel model. Hence, on average, IEEE 802.11 channel model performs 1.1m better than the RT channel model. The mean error, error variance and the Root-Mean-Square-Error (RMSE) for the two channel models are presented in Table 3.

Table 3: Comparison of mean error, error variance and RMSE error for Ray Tracing and IEEE 802.11 channel models using 1 AP, 8TPs and 24 sample points

	Mean Error (m)	Error Variance (m^2)	RMSE Error (m)
Ray Tracing channel model	5.9	16.3	7.2
Site-specific IEEE 802.11 channel model	4.8	10.4	5.8

The Cumulative Distribution Function (CDF) of the error probabilities for both the RT and the IEEE 802.11 channel models is plotted in Figure 18.

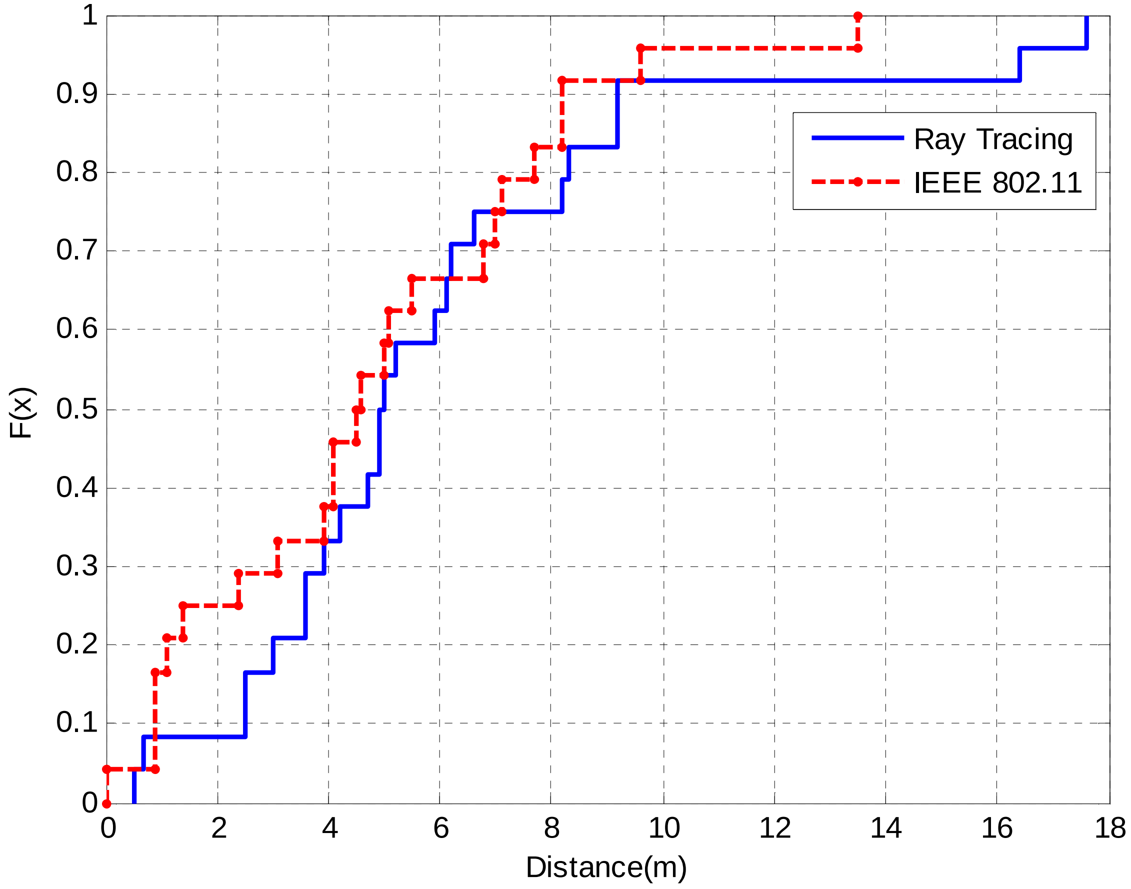


Figure 18: CDF of localization error for Ray Tracing and site-specific IEEE 802.11 channel models

5.2 Performance Evaluation of WIFI RFID Localization Technologies

In this section, the localization performance of two different WiFi RFID tags is evaluated. These WiFi RFID tags are T201 WiFi RFID tags manufactured by Ekahau. The two tags are different on the basis that one tag is an old design while the other tag is an updated design of the same model. The testbed configuration similar to the one shown in Figure 13 was used for this analysis.

The experiment was conducted for the third floor scenario of the Atwater Kent building at Worcester Polytechnic Institute. The site-specific IEEE 802.11 channel model was used by Propsim to simulate the channel between the access points and the WiFi RFID tag. The Ekahau software was trained at eight different locations on the floor plan. The localization performance of the RFID tag was then evaluated by calculating the mean error between the actual location and the location predicted by the Ekahau software. This experiment was then repeated for the second tag. Figure 19 shows the location of the access points and the training points used in this experiment.

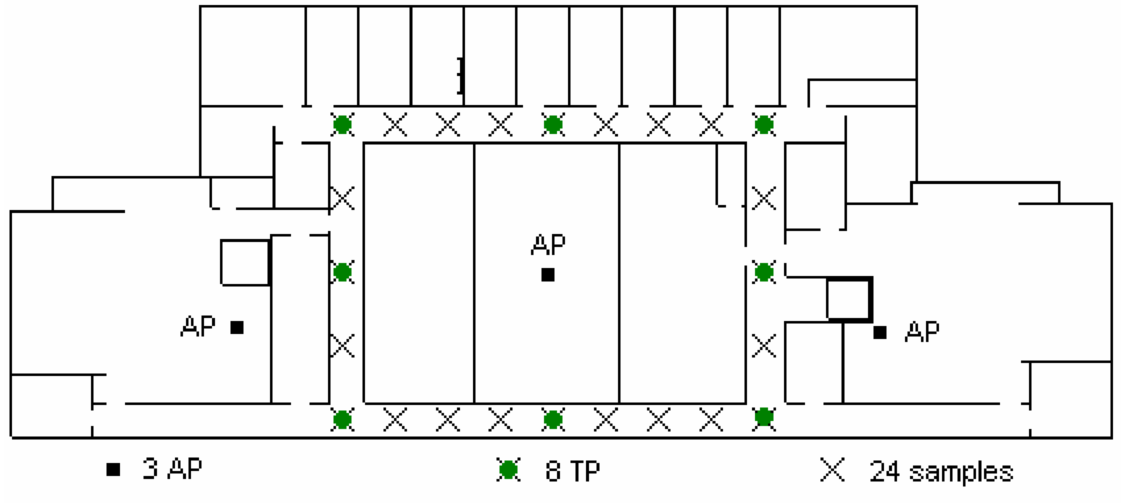


Figure 19: Configuration of access points and training points for comparing localization performance of WiFi RFID tags

In this experiment, it was found that the mean localization error is 2.8m for RFID Tag-033625 and 3.8m for RFID Tag-012970. Hence, on average, Tag-033625 performs 0.2m better and also has less error variance as compared to Tag-012970. The mean error, error variance and the RMSE for the two RFID tags are presented in Table 4.

Table 4: Comparison of mean error, error variance and RMSE error for two different WiFi RFID tags using 1 AP, 8TPs and 24 sample points

	Mean Error (m)	Error Variance (m^2)	RMSE Error (m)
TAG-033625	2.8	5.1	3.6
TAG-012970	3.0	8.0	4.1

The CDF of error probabilities for the two RFID tags is plotted in Figure 20.

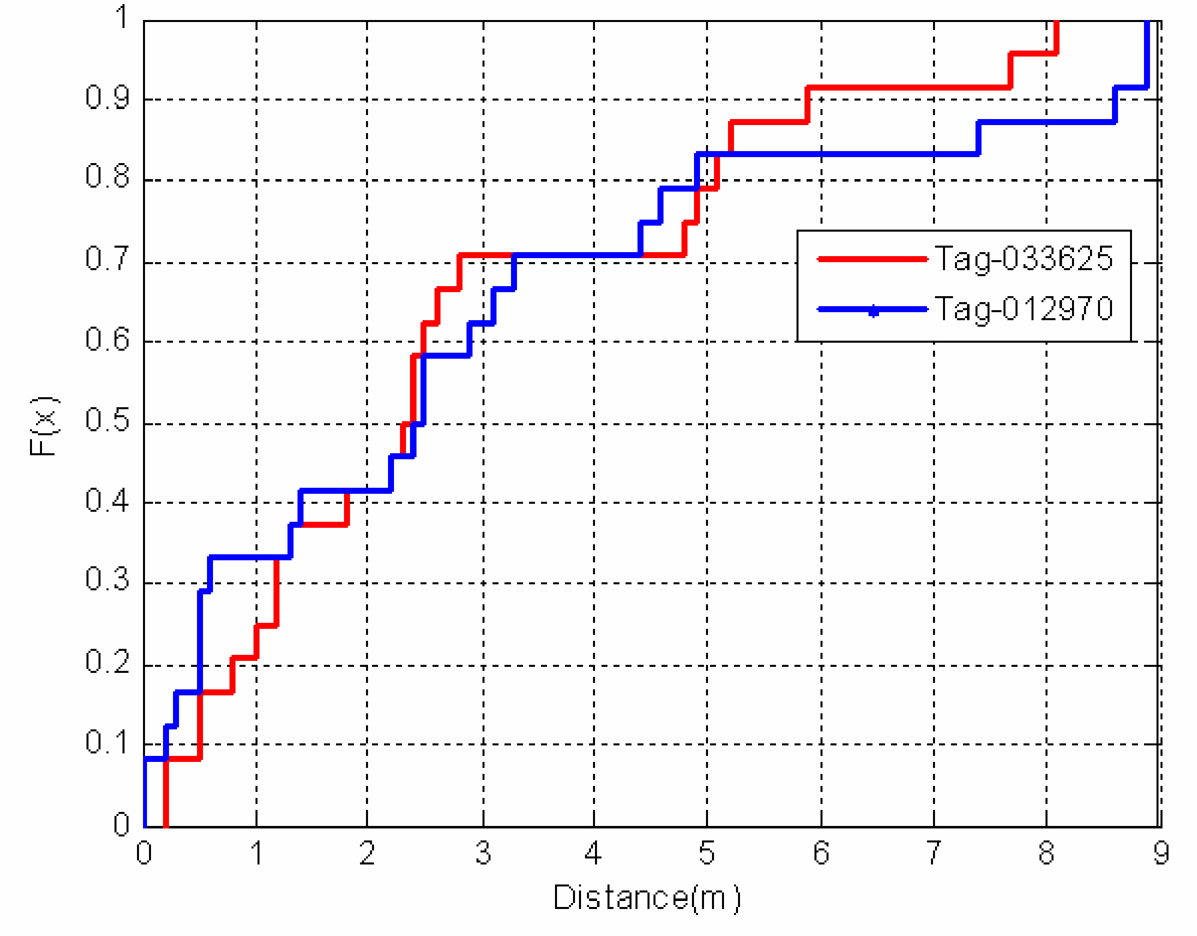


Figure 20: CDF of localization error for two different RFID tags

Chapter 6: Conclusions and Future Work

This chapter is divided into two sections. Section 6.1 includes the conclusions drawn from this research. Section 6.2 presents some of the proposed future work that can be extended from this research.

6.1 Conclusions

In this thesis, a framework was presented for comparative performance evaluation of WiFi RFID technologies in a controlled and repeatable laboratory environment. The testbed was designed using actual wireless hardware including wireless access points, real-time channel emulator hardware, active RFID modules and a laptop with a PCMCIA wireless card, all physically connected to each other. The channel model between the wireless access points and the active RFID module is simulated using the site-specific IEEE 802.11 channel model developed during this research. The site-specific IEEE 802.11 channel model was developed by combining the spatial correlation properties of shadow fading with the standard IEEE 802.11 channel model.

The testbed designed in this research was then used for comparative evaluation of localization performance of two channel models and two different WiFi RFID devices. The performance evaluation results of the two channel models, ray tracing channel model and IEEE 802.11 channel model, showed that if the IEEE 802.11 channel model is implemented with spatially correlated shadow fading, it will provide localization performance similar to that of the ray tracing channel model. This is an important result since the IEEE 802.11 channel model is not only easy to implement but also requires minimal computational resources. Ray tracing channel model, on the other hand, is hard to implement and requires extensive computational resources and a third party software.

The two different WiFi RFID tags that were compared using the testbed are developed by the same manufacturer but are using different positioning algorithms. The performance evaluation results obtained from the testbed clearly showed that one of the WiFi RFID tag performed better than the other RFID tag. These results verify the proof of concept of the testbed. Since the testbed can be used to repeat the same experiment without changing any parameters, it provides a performance benchmark that can be used to compare different RFID devices.

6.2 Future Work

In this thesis, the testbed was used for comparing only WiFi RFID devices. These RFID devices use RSS localization technique for locating and tracking purposes. It will be good to compare the performance of other RFID devices, such as UWB RFID that are based on TOA localization technique. The results obtained from comparative performance evaluation of these different RFID technologies can be used to first identify performance bottlenecks in each technology and later on to investigate improvement strategies.

The testbed design presented in this thesis is limited for RFID technologies that have bandwidths of less than 70 MHz. This bandwidth limitation is caused by the real-time channel emulator hardware used to simulate the wireless channel between the reference points and the RFID devices. This bandwidth limitation could prevent the testbed from being used for TOA-based RFID technologies which have wider bandwidth requirements. Further research can be done to replace the channel emulator hardware with the one that can support larger bandwidths. In addition, channel modeling techniques can be investigated to limit the bandwidth requirements of TOA based RFID technologies, if no proper replacements for the channel emulator hardware can be found.

Furthermore, a technology benchmark could also be created by studying the interference of different RFID technologies in a coexisting environment using collected experimental data as well as theoretical channel models for RSS and TOA localization techniques. This could eventually lead to applications that could be used to demonstrate the real-time interoperation of different RFID technologies.

Appendix A: Using Ekahau WiFi Tag T201 with EPE 3.1

This appendix outlines the details of a demo setup for using the Ekahau T201 WiFi tag with the Ekahau Real-Time Location System (RTLS). "The Ekahau Real-Time Location System is a wireless radio frequency solution that continually monitors and reports real-time locations of tracked resources [32]". Figure 21 provides an overview of the different components that can be used with Ekahau RTLS:

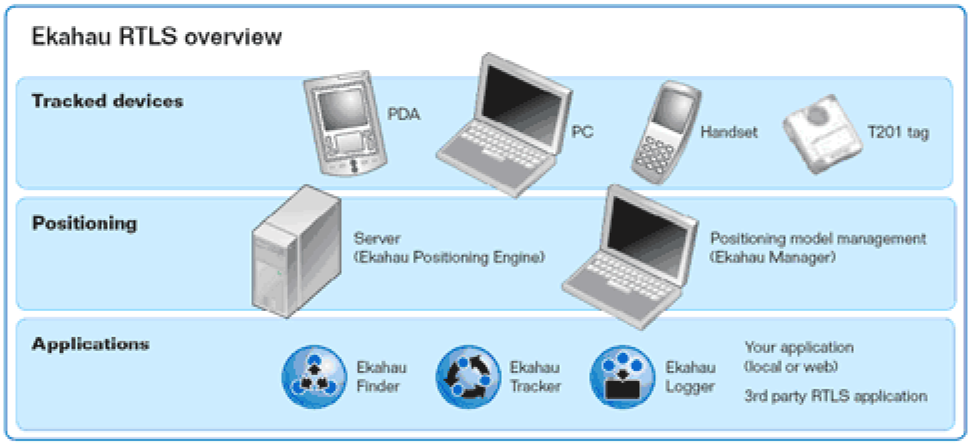


Figure 21: Overview of Ekahau RTLS Technology [32]

This demo setup uses the following components:

- Laptop with Ekahau products installed on it
    - Ekahau Positioning Engine 3.1 or higher
    - Ekahau Client 3.3 or higher

- WiFi Access Point based on 802.11 b/g technology
    - ScanTool software for access point setup
- Ekahau WiFi Tag T201
- Cross-over Ethernet Cable
- USB-to-serial adaptor cable (if the laptop does not have a serial port)
    - Drivers from manufacturer website installed on the laptop
- T201 power cable with the serial interface

This appendix is divided into different sections to facilitate the setup process. By the end of this appendix, all the setup details for the different components of the demo setup will be revealed. The same setup procedure is required for using these components with the real-time laboratory testbed design presented in this thesis.

A.1 Software Installations

This demo setup consists of only two main hardware components, the laptop and the RFID tag. The rest of the components are software applications. This section highlights the necessary steps needed to successfully install these software applications.

A.1.1 Ekahau Positioning Engine Installation

The ‘epe3.1’ folder contains the necessary installation files for installing Ekahau Positioning Engine (EPE) 3.1. The following steps can be used to install EPE 3.1 on the laptop:

- Open the ‘epe 3.1’ folder.
- Open the ‘EPE3.1_Installation_File’ folder.
- Double-click on the ‘Install_Engine_win32’ file to start the setup wizard.
- Choose the ‘Full’ installation option
- Install the following features:
    - Engine 3.1
    - Manager 3.1
    - Drivers
    - Client for Win32

- Follow the instructions on the screen until the installation process is completed.

A.1.2 ScanTool Software Installation

The ScanTool software is used to detect the Orinoco AP2000 access points connected to the Ethernet port of the laptop. The setup files are included in the folder 'ScanTool'.

- Open the 'ScanTool' folder.
- Double-click on the Setup installation file, this will start the setup wizard.
- Follow the instructions in the wizard.

This will install the ScanTool program, the setup will put a shortcut to this program on Windows desktop. Alternately, the ScanTool.exe file present in the 'ScanTool' folder can also be used to run this program.

A.1.3 Serial-USB Adaptor Driver Installation

The Serial-USB adaptor cable used for this setup is Belkin F5U409. The system files required for installing this device are included in the folder 'Belkin_USB_Serial_F5U409'.

- Connect the adaptor cable to the USB port on the laptop.
- Windows will detect the USB device and will initiate the driver installation wizard.
- Provide the location of the installation folder 'Belkin_USB_Serial_F5U409'.

This will install the drivers for the adaptor cable. Verify that the installation was successful. Follow the following verification steps:

- Go to ‘Control Panel’ in Windows.
- Double-click on ‘System’, this will open a ‘System Properties’ window.
- Select the ‘Hardware’ tab.
- Click on the ‘Device Manager’ button, this will open a ‘Device Manager’ window.
- Expand the ‘Ports (COM & LPT)’ node form the tree view.
- Verify that ‘Belkin Serial On USB Port’ tree sub-node is present.

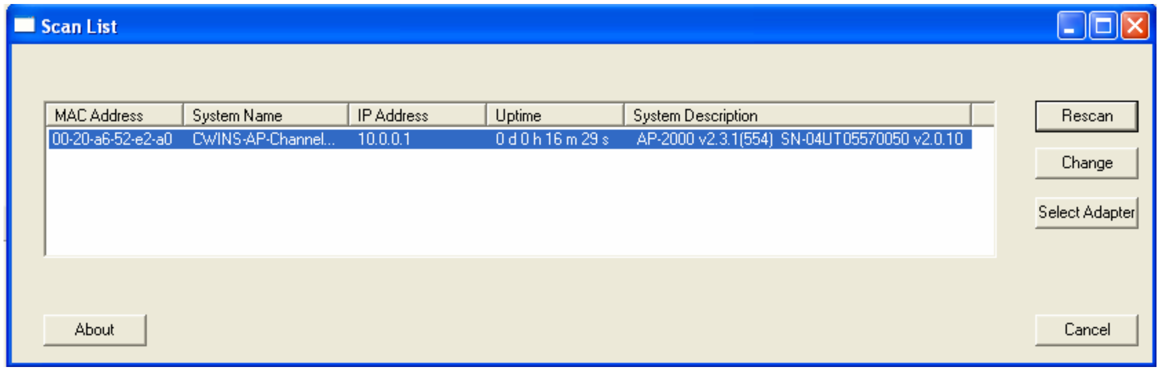
A.2 Orinoco AP2000 Access Point Setup

The Orinoco AP2000 access point was used for this demo. The AP2000 needs to be configured with a DHCP option and its wireless interface needs to be set up appropriately. The following steps can facilitate the setup process:

- Insert a supported wireless card in ‘PC Card A’ slot of the access point.
- Connect the access point to the Ethernet port of your laptop using a cross-over Ethernet cable.

- Power up the access point, and wait for it to reboot. When the LEDs on the access point stop flashing and the power LED becomes ‘Green’, the access point has been powered up. This process usually takes 10-15 seconds.

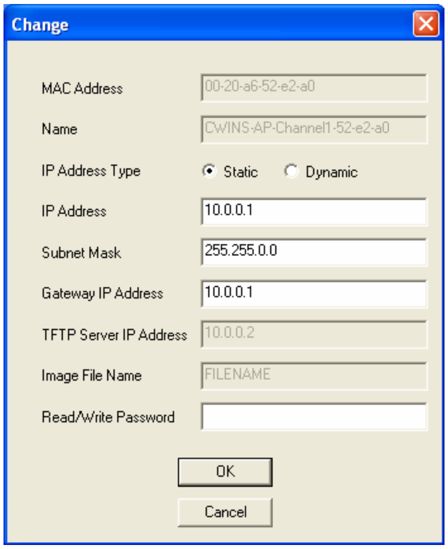
- Start the ScanTool program; you will see the name of your access point as the program scans for the connected AP2000 access points. If you do not see any access point listed, click on the Rescan button. If the access point is still not located, make sure that the windows firewall and other firewall softwares are turned off. Figure 22 shows the ScanTool user interface:



MAC Address	System Name	IP Address	Uptime	System Description
00-20-a6-52-e2-a0	CWINS-AP-Channel...	10.0.0.1	0 d 0 h 16 m 29 s	AP-2000 v2.3.1(554) SN-04UT05570050 v2.0.10

Figure 22: ScanTool User Interface

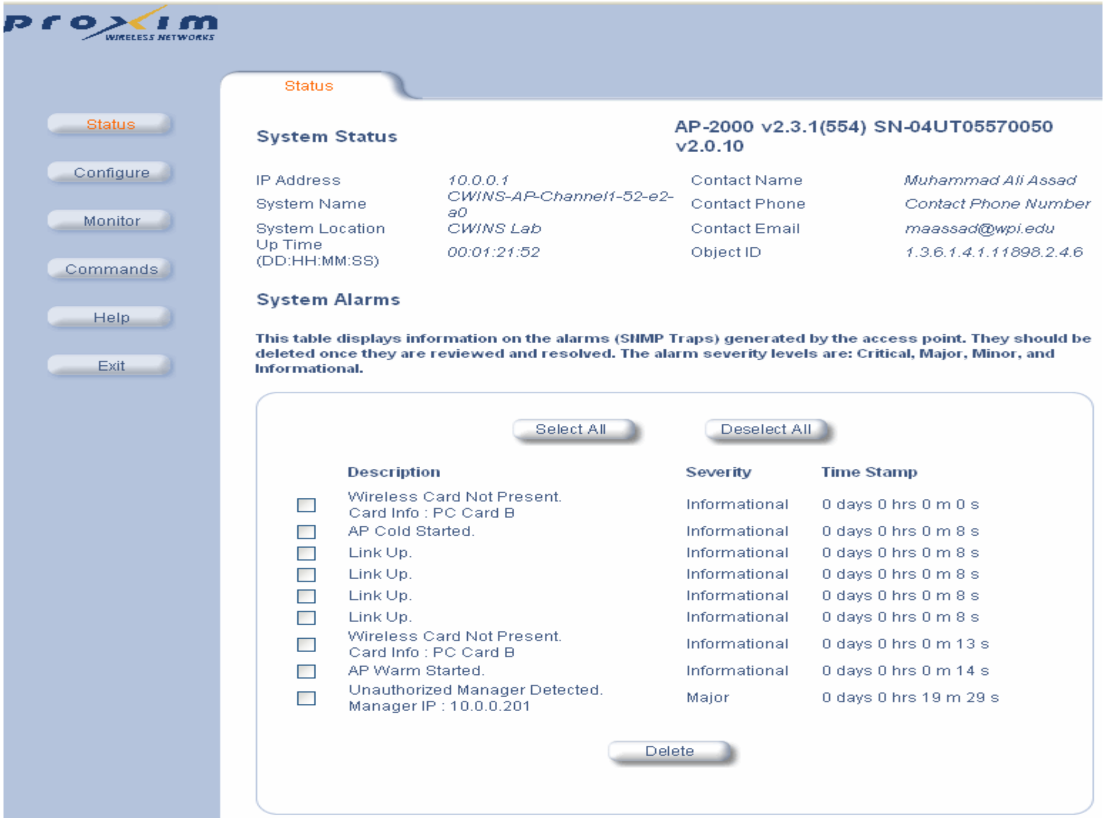
- Make sure that the IP Address of the access point has the same subnet as the IP address of the laptop’s Ethernet adaptor. If not change it by clicking on the ‘Change’ button on the ScanTool user interface. Figure 23 shows the user interface for changing the IP address of the access point.



MAC Address	00-20-a6-52-e2-a0
Name	CWINS-AP-Channel1-52-e2-a0
IP Address Type	● Static ○ Dynamic
IP Address	10.0.0.1
Subnet Mask	255.255.0.0
Gateway IP Address	10.0.0.1
TFTP Server IP Address	10.0.0.2
Image File Name	FILENAME
Read/Write Password	

Figure 23: User Interface for changing AP2000 IP address

- Enter the IP address of the access point in the internet browser to open the HTTP interface of the AP2000 access point. The user interface of the AP2000 is shown in Figure 24.



System Status

AP-2000 v2.3.1(554) SN-04UT05570050 v2.0.10

IP Address	*10.0.0.1*	Contact Name	*Muhammad Ali Assad*
System Name	*CWINS-AP-Channel1-52-e2-a0*	Contact Phone	*Contact Phone Number*
System Location	*CWINS Lab*	Contact Email	*maassad@wpi.edu*
Up Time (DD:HH:MM:SS)	*00:01:21:52*	Object ID	*1.3.6.1.4.1.11898.2.4.6*

System Alarms

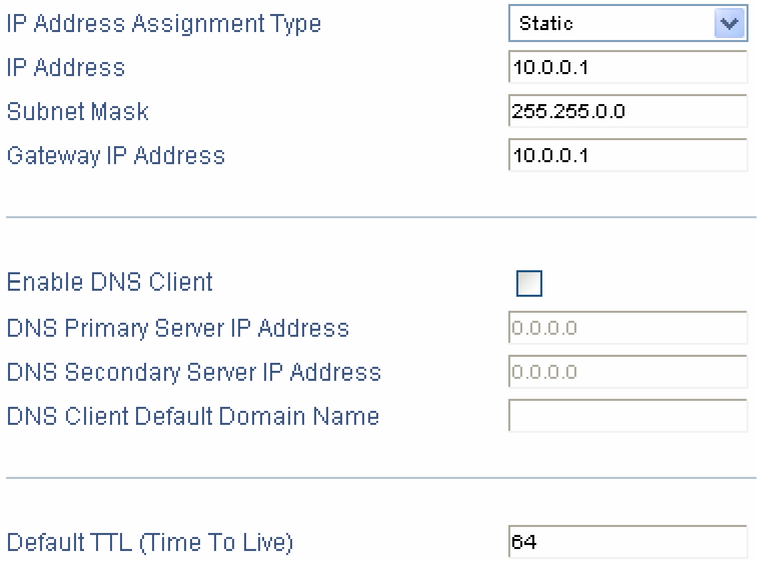
This table displays information on the alarms (SNMP Traps) generated by the access point. They should be deleted once they are reviewed and resolved. The alarm severity levels are: Critical, Major, Minor, and Informational.

	Description	Severity	Time Stamp
□	Wireless Card Not Present. Card Info : PC Card B	Informational	0 days 0 hrs 0 m 0 s
□	AP Cold Started.	Informational	0 days 0 hrs 0 m 8 s
□	Link Up.	Informational	0 days 0 hrs 0 m 8 s
□	Link Up.	Informational	0 days 0 hrs 0 m 8 s
□	Link Up.	Informational	0 days 0 hrs 0 m 8 s
□	Link Up.	Informational	0 days 0 hrs 0 m 8 s
□	Wireless Card Not Present. Card Info : PC Card B	Informational	0 days 0 hrs 0 m 13 s
□	AP Warm Started.	Informational	0 days 0 hrs 0 m 14 s
□	Unauthorized Manager Detected. Manager IP : 10.0.0.201	Major	0 days 0 hrs 19 m 29 s

Figure 24: User Interface for AP2000 setup

- Click on the ‘Configure’ button and select the ‘Network’ tab.

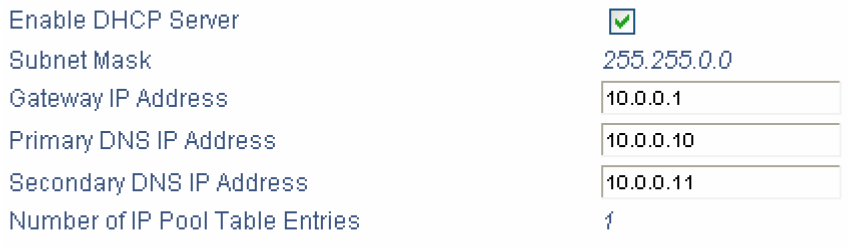
- Select the ‘IP Configuration’ tab and set up the fields as shown in Figure 25.



IP Address Assignment Type	Static
IP Address	10.0.0.1
Subnet Mask	255.255.0.0
Gateway IP Address	10.0.0.1
Enable DNS Client	□
DNS Primary Server IP Address	0.0.0.0
DNS Secondary Server IP Address	0.0.0.0
DNS Client Default Domain Name	
Default TTL (Time To Live)	64

Figure 25: IP configuration for the AP2000

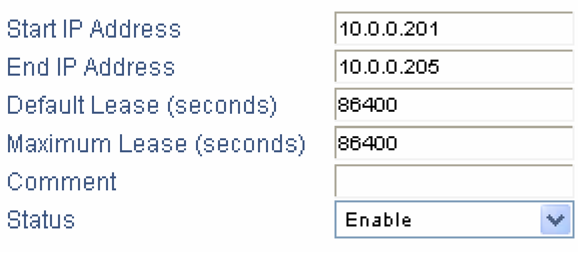
- Select the ‘DHCP Server’ tab and set up the fields as shown in Figure 26.



Enable DHCP Server	☑
Subnet Mask	255.255.0.0
Gateway IP Address	10.0.0.1
Primary DNS IP Address	10.0.0.10
Secondary DNS IP Address	10.0.0.11
Number of IP Pool Table Entries	1

Figure 26: DHCP server configuration for AP2000

- Click on the ‘Add’ button in the IP Pool Table panel and set up the fields as shown in Figure 27.

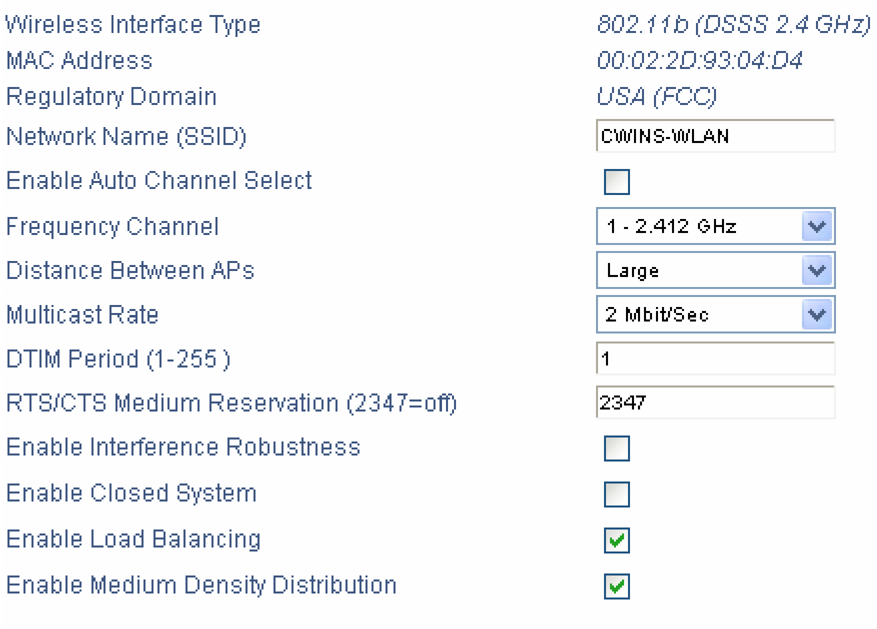


Start IP Address	10.0.0.201
End IP Address	10.0.0.205
Default Lease (seconds)	86400
Maximum Lease (seconds)	86400
Comment	
Status	Enable

Figure 27: IP pool table configuration for AP2000

- Click on the ‘Interfaces’ tab in the ‘Configure’ window.

- Select ‘Wireless-A’ tab and set it up as shown in Figure 28. If multiple access points are used, choose a different frequency channel for each access point, preferably 1, 6 or 11.



Wireless Interface Type	*802.11b (DSSS 2.4 GHz)*
MAC Address	*00:02:2D:93:04:D4*
Regulatory Domain	*USA (FCC)*
Network Name (SSID)	CWINS-WLAN
Enable Auto Channel Select	□
Frequency Channel	1 - 2.412 GHz
Distance Between APs	Large
Multicast Rate	2 Mbit/Sec
DTIM Period (1-255)	1
RTS/CTS Medium Reservation (2347=off)	2347
Enable Interference Robustness	□
Enable Closed System	□
Enable Load Balancing	☑
Enable Medium Density Distribution	☑

Figure 28: Wireless interface setup for AP2000

- Reboot the access point after completing these settings. The Ethernet cable can now be disconnected from the laptop.

- After the access point is rebooted, connect the laptop’s wireless adaptor to the ‘CWINS-WLAN’ wireless network. Obtain the IP address of the laptop after it is connected to the wireless network. Since the laptop is the only wireless network client connected to the network, it will be assigned.

A.3 Ekahau WiFi RFID T201 Setup

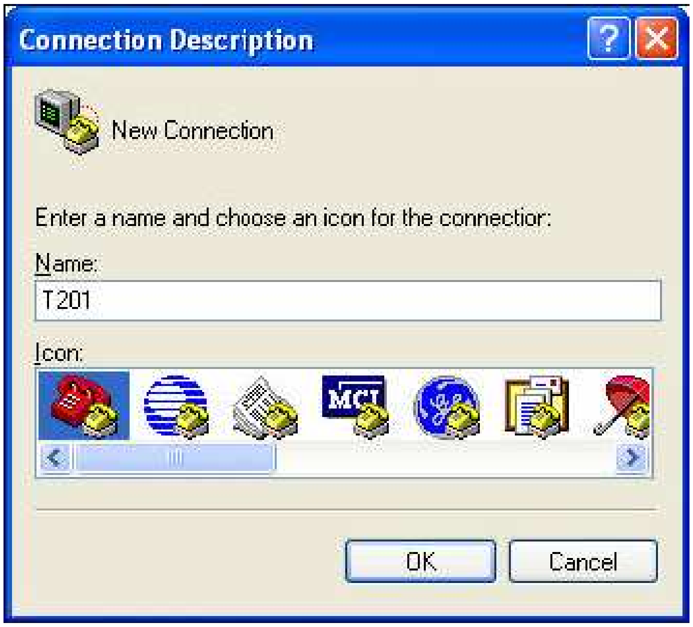
The Ekahau T201 WiFi tag needs to be configured before it can be used with the access point and the EPE 3.1. The following steps can be used to connect to the tag and configure its properties:

- Connect the tag to the power cable. The power cable connects to the tag using the USB interface. The power cable not only provides power to the tag and charges the internal battery but also provides a serial interface to configure its properties.

- Connect the serial interface of the cable to the serial port of a computer. This can be done using a serial-to-serial cable. Alternatively, the serial interface of the cable can also be connected to the USB interface of the computer using a serial-to-USB interface cable.

- The tag supports a Command Line Interface (CLI) interface which is what we are going to use to connect to the tag.

- Open any terminal program, such as Windows HyperTerminal. Click **Start > Programs > Accessories > Communications >HyperTerminal**. The dialog box shown in Figure 29 will pop up.



Connection Description

New Connection

Enter a name and choose an icon for the connection:

Name:

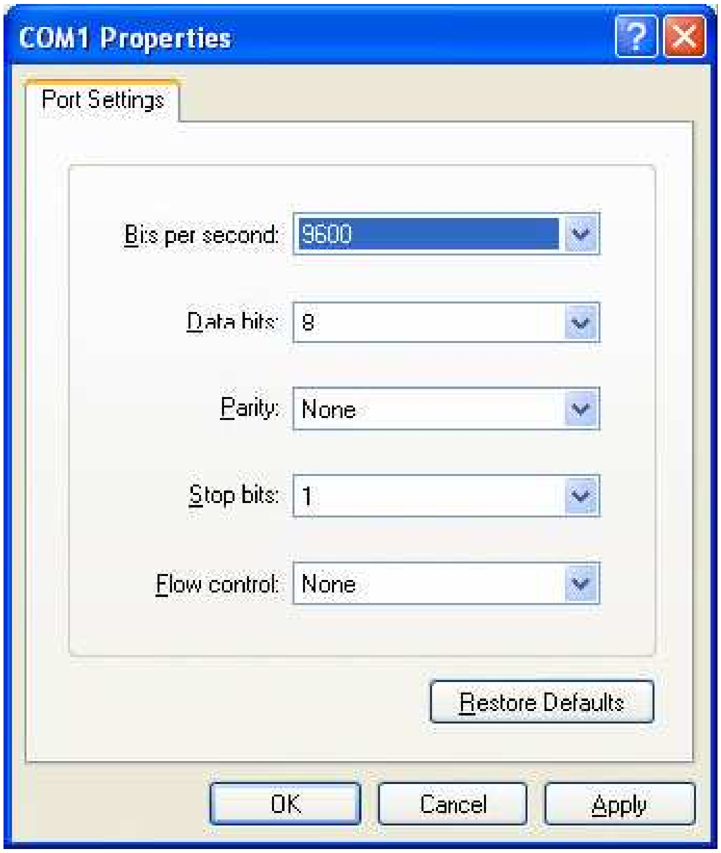
T201

Icon:

Figure 29: Dialog box to name the connection to the T201 RFID tag using HyperTerminal

- Provide an appropriate name to the connection such as ‘T201’. Connect using a COM port; provide the appropriate port number to which the cable was connected in the previous step.

- Set up the COM port settings as shown in Figure 30:



Bits per second: 9600

Data bits: 8

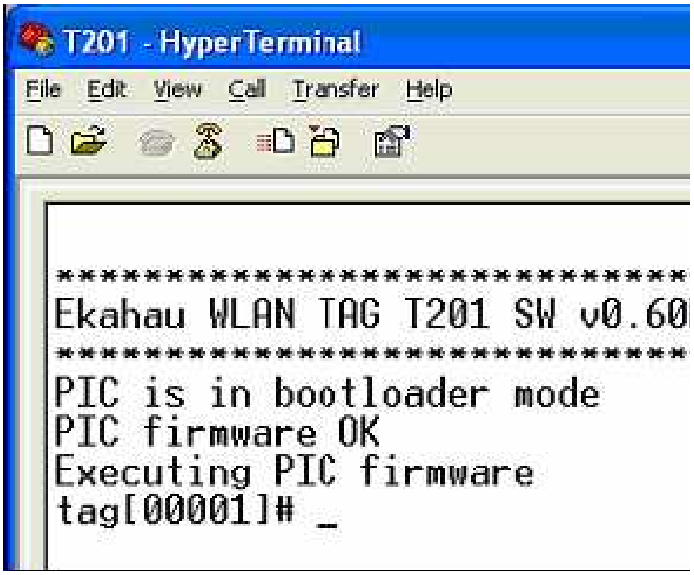
Parity: None

Stop bits: 1

Flow control: None

Figure 30: COM port settings for the HyperTerminal connection

- Switch ON the RFID tag by pressing the RED power button on the tag and holding it for 5 seconds. The CLI prompt will show the output as shown in Figure 31 when the RFID tag is turned on:



```
******************************
Ekahau WLAN TAG T201 SW v0.60
******************************
PIC is in bootloader mode
PIC firmware OK
Executing PIC firmware
tag[00001]# _
```

Figure 31: CLI prompt output upon connection to the tag

- To allow T201 tag to associate with the WiFi access points, at least the following parameters need to be configured:
  - SSID of the AP
  - Positioning Engine IP address
  - Disable WEP key
  - Ensure that the IP method is set to DHCP
  - T201 friendly name

  When these properties are configured, the ‘save’ command needs to be executed to save the configurations in the memory of the RFID tag.

- An example of the usage of these commands is shown below:

```
tag[00001]# wlan ssid 1 “CWINS-WLAN”<Enter>

tag[00001]# wlan wep 1 “off”<Enter>

tag[00001]# engine ip 10.0.0.201<Enter>

tag[00001]# ip method <Enter>   (verify that the ip method is DHCP)

tag[00001]# name CWINS-RFID <Enter>

tag[00001]# save<Enter>
```

- The RFID with the current configuration will scan the network on the following action events:
    - Periodic Scan (every 120 s)
        - To configure this property, type: *periodic active <1 or 0>*
        - To change the periodic wakeup interval, type: *periodic wakeup interval <time in seconds>*
        - To change the scan count, type: *periodic scan count <number>*
    - Motion event (when the tag is moved)
        - To configure this property, type: *motion active <1 or 0>*
        - To change the scan count, type: *motion scan count <number>*
    - Aftermotion event (after the tag has stopped moving)

        - To configure this property, type: *aftermotion active <1 or 0>*

        - To change the scan count, type: *aftermotion scan count <number>*

    - Button event (after the button on the tag is pressed)

        - To configure this property, type: *button active <yes or no>*

        - To change the scan count, type: *button scan count <number>*

- The status of the tag can be viewed by typing *status* command

A.4 Ekahau Manager Setup

Ekahau Manager is the Graphical User Interface used for training and utilizing the Ekahau positioning engine. The Ekahau positioning engine 3.1 is required to access the T201 WiFi RFID tag. The steps required for configuring the Ekahau manager to track the RFID tag are discussed below:

- Open the Ekahau Client Controller by double-clicking on the shortcut icon from **Start > Programs > Ekahau** menu.

- Open the Ekahau Manager by double-clicking on the shortcut icon from **Start > Programs > Ekahau** menu.

- When the Ekahau manager is loaded, open a new positioning model.

- Load the floor plan.

- Plot the rails/paths along which the user/asset is going to move during the training procedure. Please follow the instructions provided in the Ekahau User Guide.

- Now, click on the Devices tab at the bottom of the GUI. Verify that the RFID device is listed along with the localhost device. If it is not listed, please ensure that the RFID tag is properly configured and have the correct engine ip address for reference. Also, make sure that your PC/laptop is connected to the wireless network (SSID) to which the RFID is going to connect. If everything is set up properly, also verify if the tag is turned on. Also, ensure that the tag displays GREEN light when it scans. If it is displaying red light, there has to be some error in its configuration or it is not able to connect to the access point or it can not find the positioning engine at the specified ip address. If all the settings are correct, then exit the Ekahau client and the Ekahau manager and load them again. One of these actions shall resolve the issue.

- Now select RFID tag as your preference for the training procedure by selecting the RFID device in the dialog box that appears when the **File > Preferences** menu item is selected.

- Now, click on the Signal Strength tab at the bottom of the GUI and verify whether the positioning engine is receiving any RSS information from the RFID tag.

- Now, train the positioning model at a number of points on the rails/paths defined earlier. For instructions and help with this procedure, refer to the Ekahau User Guide.

- After training, load the positioning model into the positioning engine by selecting the appropriate menu item from the **File** menu.
- Now, you can track the RFID device by selecting the ‘tracking’ check box next to the name of the RFID tag from the Devices tab.
- You can move around the floor with the tag attached to your clothes, and the Ekahau positioning engine with track you.
- The accuracy of the localization varies with the number of access points, number of training points and the position of the access points.

Now, you are all set to track the RFID tag. ENJOY!

Appendix B: MATLAB Code

B.1 MATLAB Function for Calculating RSS with and without fading based on IEEE 802.11 Channel Model C

```
function [RSS_with_fading, RSS_without_fading] =
WLAN_channel_calculation(d)
% WLAN_channel_calculation(d): This function can be used to
calculate the
% RSS value at different distances based on the IEEE 802.11
channel model
% C.  For other channel models, the simulation parameters
can be changed
% appropriately.  'd' can be supplied both as a single
value or an array.
% The output provides the calculation of RSS with and
without fading.
%
% Author: Muhammad Ali Assad

%% IEEE 802.11b Model C Simulation Parameters
Pt = -8;  % transmit power (dBm)
% First meter loss for indoor enviornment (measurement
based)
Lo = 31;
db = 5; % break-point distance (m)
% distance power gradient till breakpoint
a1 = 2;
% distance power gradient after breakpoint
a2 = 3.5;
% shadow fading standard deviation till breakppoint
sigma1 = 3;
```

```
% shadow fading standard deviation after breakppoint
sigma2 = 5;

%% Calculation here
% With fading
for i=1:numel(d)
    if(d(i) <= db)
        %RSS_with_fading(i) = Pt - Lo - 10*a1*log10(d(i)) + 
normrnd(0,sigma1);
        RSS_with_fading(i) = Pt - Lo - 10*a1*log10(d(i)) + 
sigma1*randn(1);
    else
        %RSS_with_fading(i) = Pt - Lo - 10*a1*log10(db) - 
10*a2*log10(d(i)/db) + normrnd(0,sigma2);
        RSS_with_fading(i) = Pt - Lo - 10*a1*log10(db) - 
10*a2*log10(d(i)/db) + sigma2*randn(1);
    end
end
% Without fading
for i=1:numel(d)
    if(d(i) <= db)
        RSS_without_fading(i) = Pt - Lo - 
10*a1*log10(d(i));
    else
        RSS_without_fading(i) = Pt - Lo - 10*a1*log10(db) - 
10*a2*log10(d(i)/db);
    end
end
```

B.2 MATLAB Function for Extracting RSS and Distance Information From a Ray Tracing Generated .RAW File

```
function [distance,norm_power_dB] =
rayTracingChanInfo(fName,dispChannel)
% RAYTRACINGCHANINFO extracts and displays the channel
information from a
%   RayTracing generated .RAW file.  It can also be used to
generate the
%   channel impulse response.
%
% INPUT:
% ------
% fName : Filename of the .RAW file (including the
extension .raw)
% dispChannel : Displays the channel impulse response if
true, else only
%                returns the channel info if false.
%
% OUTPUT:
% -------
% distance: Returns the distance between the TX and the RX
% norm_power_dB: Returns the Received Power at RX in dB
%
% Author: Muhammad Ali Assad, Mohammad Heidari
% Last Modified: 02/24/2007

%% Reading the .RAW file and extracting the required data
%
filename = fName;

%------ read the whole file in one string ----------------
```

```
fid = fopen(filename, 'r');
  s1 = fscanf(fid, '%c');
fclose(fid);

%------ remove the header characters ----------------------
k = 1;
while s1(k) ~= '_'
   k = k + 1;
end;
while s1(k) == '_'
   k = k + 1;
end;

j = 0;
for i=k:length(s1)
   j = j + 1;
   s2(j) = s1(i);
end;

% Complex coefficient
jay = sqrt(-1);

%------ convert to numbers and sort -----------------------
s3 = str2num(s2);
rt = sortrows(s3,1);

tk  = rt(:,1);              % time values in seconds
ak  = rt(:,2);              % amplitude values (linear/no
units)
phk = rt(:,3);              % phase values (in radians)
dep_angle = rt(:,4);        % departure angle values (in
radians)
```

```
arr_angle = rt(:,5);        % arrival angle values (in
radians)
n_cross   = rt(:,6);        % number of crossings
n_ref     = rt(:,7);        % number of reflections

%% Modifying/Manipulating the data for best results
%
npaths = length(tk);                % total number of taps
ck     = ak .* exp(-jay * phk);     % construct the complex
taps
ck_db  = 20 * log10(abs(ck));       % tap amplitude values
in dB

% --------------------- Purely for making nice plots ------
---------------
tmin = 10*(floor(1e9*tk(1)/10)-1)*1e-9;
tmax = 10*ceil(1e9*tk(npaths)/10)*1e-9;

min_ckdb = min(ck_db);
min_db1  = 10*floor(min_ckdb/10);
min_db   = max(min_db1, -100);

max_ckdb = max(ck_db);
max_db   = 10*ceil(max_ckdb/10);

%% Calculating and generating the channel information
%
%----------------------------------------------------------
---------------
% --------------------------- Computing Power -------------
---------------
point_information = filename(1:strfind(filename,'_')-1);
ak2 = ak .* ak;
norm_power = ones(1, npaths)*ak2;
```

```
norm_power_dB = 10*log10(norm_power);

%----------------------------------------------------------
---------------
% ------------------------- Computing Distance ------------
---------------
[txx, txy, rxx, rxy, distance] = textread(filename,...
    'Tx = ( %d,  %d) Rx = ( %d,  %d)  Dist =  %f 
meters.',1);

if (dispChannel)
    %------------------------------------------------------
----------------
    %----------------------- Computing Dealy Spread -------
----------------
    tk2 = tk .* tk;
    rms1 = (tk2' * ak2) ./ norm_power;
    rms2 = ((tk' * ak2) ./ norm_power) .^ 2;
    trms = (rms1 - rms2) .^ 0.5

    ex_delay = max(tk) - min(tk);

    %------------------------------------------------------
----------------
    %-------------------- Impulse Response dB Scale Plot --
----------------
    figure;
    for i=1:npaths
        hold on
        plot([tk(i) tk(i)] * 1e9, [min_db ck_db(i)]);
    end;
    axp = [tmin*1e9 tmax*1e9 min_db max_db];
    axis(axp);
    title('Impulse response in dB scale')
    xlabel('time in ns')
    ylabel('dB')

    % Note: axp, and axis(axp) are just for keeping the
plot within bounds
    % defined by tmin, tmax, min_db and max_db

    %-------------------------------------------------------
----------------
    %------------------ Impulse Response Linear Scale Plot
----------------
    figure;
    for i=1:npaths
        hold on
        plot([tk(i) tk(i)] * 1e9, [0 abs(ck(i))*1000]);
    end;
    axp = [tmin*1e9 tmax*1e9 0 max(abs(ck))*1.1*1000];
    axis(axp);
    title('Impulse response in linear scale')
    xlabel('time in ns')
    ylabel('mU') % mU stands for milli Units
end
```

B.3 MATLAB Code for Displaying RSS and Distance Information from All the .RAW Files Present in the Same Directory

```
% File: rayTracingLoop.m
% This code can be used to extract the RSS and distance
information from all
% the .RAW files present in the same directory.  The output
is generated in a
% three-column format.  The first column displays the point
information, the
% second column represents the distance from the access
point at that point,
% and the third point represents the RSS at that point.
%
% Author: Muhammad Ali Assad

RT_files = dir('*.raw');
filenames = {RT_files.name};
nPoints = length(filenames);
distance(1:nPoints) = 0;
RT_RSS(1:nPoints) = 0;
point(1:nPoints) = 0;
for i = 1:nPoints
    [distance(i),RT_RSS(i)] =
rayTracingChanInfo(filenames{i},false);
    point(i) =
str2double(filenames{i}(strfind(filenames{i},'_')+ ...
        1:strfind(filenames{i},'.')-1)) + 1;
end
veri = [point' distance' RT_RSS'];
verisort = sortrows(veri,1)
```

B.4 MATLAB Code for Plotting RSS of Ray Tracing Channel Model and IEEE 802.11 Channel Model without Fading, With Fading and With Correlated Fading

```
% File: WLAN_vs_RT.m
% This code plots the RSS information of Ray Tracing
channel model and the
% IEEE 802.11 channel model without fading, with fading and
with spatially
% correlated fading.
%
% Author: Muhammad Ali Assad

%% IMPORT DATA
% The data should be arranged in three column format.  The
first column
% should have the point information, the second column
should have the
% corresponding distance information between the point and
the access
% point, and the third column should have the RSS
information.  The
% MATLAB code in rayTracingLoop.m generates the output in
the correct
% format.

% load RSS_MH_1AP_loopdata
% load RSS_1AP_8TP_Mid
load RSS_1AP_8TP_Cyg
distance_point = RSS_dist_array(:,2);
nPoints = length(distance_point);
RT_RSS = -8 + RSS_dist_array(:,3);
[WLAN_RSS_fading, WLAN_RSS] =
WLAN_channel_calculation(distance_point);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%% Spatial correlation code
sig = 5;
fading = zeros(1,nPoints);
fading(1) = sig*randn(1);
% Chosen based on the best smooth fit between RT and WLAN
RSS data
corr_coeff = 0.956;
for i=2:nPoints
    fading(i) = corr_coeff*fading(i-1) + sqrt(1-
corr_coeff^2)*sig*randn(1);
end

WLAN_RSS_spatial_fading = WLAN_RSS + fading;

%% Mean error between the RSS data and WLAN spatially
correlated shadow
%% fading data
error_RSS_WLAN = WLAN_RSS_spatial_fading' - RT_RSS;
square_error = error_RSS_WLAN.^2;
mean_square_error = mean(square_error);
RMSE_RSS_WLAN = sqrt(mean_square_error);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
figure ;
subplot(3,1,1);
plot(1:nPoints,WLAN_RSS,'--bo',1:nPoints,RT_RSS,'--
rd','MarkerSize',8);
legend('IEEE 802.11 model (no fading)', 'Ray Tracing');
legend boxoff;
% xlabel('Points');
```

```
ylabel('RSS(dBm)');
% title('IEEE 802.11 Model C (without fading) and RT
channel models');
% title('RSS comparison of IEEE 802.11 Model C (without
shadow fading) and Ray Tracing channel models');
subplot(3,1,2);
plot(1:nPoints,WLAN_RSS_fading,'--bo',1:nPoints,RT_RSS,'--
rd','MarkerSize',8);
legend('IEEE 802.11 model (fading)', 'Ray Tracing');
legend boxoff;
% xlabel('Points');
ylabel('RSS(dBm)');
% title('RSS comparison of IEEE 802.11 WITH FADING and Ray
Tracing channel models');
subplot(3,1,3);
plot(1:nPoints,WLAN_RSS_spatial_fading,'--
bo',1:nPoints,RT_RSS,'--rd','MarkerSize',8);
legend('IEEE 802.11 model (correlated fading)', 'Ray
Tracing');
xlabel('Points');
ylabel('RSS(dBm)');
legend boxoff;
% plot3_title = sprintf('%s%0.3g%s','RSS comparison of IEEE
802.11 WITH CORRELATED FADING (Correlation Coefficient =
',corr_coeff,') and Ray Tracing channel models');
% title(plot3_title);
% title('RSS comparison of IEEE 802.11 Model C (with
correlated shadow fading) and Ray Tracing channel models');
title(sprintf('%s%d','IEEE 802.11 Model C (with correlated
fading) and RT channel models with RMSE =
',RMSE_RSS_WLAN));
```

B.5 MATLAB Code for plotting the Two-Piece Path Loss Model for Ray Tracing and IEEE 802.11 Channel Models

```
% File: plot_curve_fit.m
% This code plots the two piece model for both Ray Tracing
and IEEE 802.11
% channel model.  Using this code, the first meter path
loss (Lo) was
% adjusted in WLAN_channel_calculation.m to calibrate the
RSS readings with
% each other.
%
% Author: Muhammad Ali Assad

%% Import Data
% This data was extracted from the ray tracing results at
different
% distances from the Access Point
% see also: rayTracingLoop.m and WLAN_channel_calculation.m
load calibrate_data;
calibrate_data_sorted = sortrows(RT_WLAN_Calibrate,2);
RT_calibrate_distance = calibrate_data_sorted(:,2);
% Add the transmitted power of the AP to the data
RT_calibrate_power = -8 + calibrate_data_sorted(:,3);
% Calculate the corresponding RSS value for IEEE 802.11
model
[WLAN_calc_fading WLAN_calc] =
WLAN_channel_calculation(RT_calibrate_distance);

%% Break Point Line fitting with two alphas
db = 5; % Breakpoint distance
count1 = 1;
count2 = 1;
```

```
for i=1:length(RT_calibrate_distance)
    if(RT_calibrate_distance(i)<= db)
        d2less5(count1) = RT_calibrate_distance(i);
        count1 = count1 + 1;
    else
        d2more5(count2) = RT_calibrate_distance(i);
        count2 = count2 + 1;
    end
end
%%
%RT distance 5m and less
RT_log_d2less5 = 10*log10(d2less5);
RT_RSS2less5 = RT_calibrate_power(1:length(d2less5));
RT_coeffRSSless5 = polyfit(RT_log_d2less5,RT_RSS2less5',1);
RT_lineRSSless5 = polyval(RT_coeffRSSless5,0:8);
RT_alpha1 = -1*RT_coeffRSSless5(1)
RT_L0 = RT_coeffRSSless5(2)
%RT more than 5m
RT_log_d2more5 = 10*log10(d2more5);
RT_RSS2more5 =
RT_calibrate_power(length(RT_RSS2less5)+1:(length(RT_RSS2le
ss5) + length(d2more5)));
RT_coeffRSSmore5 = polyfit(RT_log_d2more5,RT_RSS2more5',1);
RT_lineRSSmore5 = polyval(RT_coeffRSSmore5,8:18);
RT_alpha2 = -1*RT_coeffRSSmore5(1)


figure (1)
subplot(2,1,1);
plot(RT_log_d2less5,RT_RSS2less5,'o',0:8,RT_lineRSSless5,'-
');
hold on
plot(RT_log_d2more5,RT_RSS2more5,'o',8:18,RT_lineRSSmore5,'
-');
```

```
RT_title = sprintf('%s\n%s%2.2f%s%1.2f%s%1.2f','RT
Calculation Best Fit Line', ...
    'Lo = ',RT_L0 + 8,'   alpha1 = ',RT_alpha1,'   alpha2 =
',RT_alpha2);
title(RT_title);
xlabel('10log10(distance)');
ylabel('RSS(dBm)');

%%
%WLAN distance 5m and less
WLAN_log_d2less5 = 10*log10(d2less5);
WLAN_RSS2less5 = WLAN_calc(1:length(d2less5));
WLAN_coeffRSSless5 =
polyfit(WLAN_log_d2less5,WLAN_RSS2less5,1);
WLAN_lineRSSless5 = polyval(WLAN_coeffRSSless5,0:7);
WLAN_alpha1 = -1*WLAN_coeffRSSless5(1)
WLAN_L0 = WLAN_coeffRSSless5(2)
%WLAN more than 5m
WLAN_log_d2more5 = 10*log10(d2more5);
WLAN_RSS2more5 =
WLAN_calc(length(WLAN_RSS2less5)+1:(length(WLAN_RSS2less5)
+ length(d2more5)));
WLAN_coeffRSSmore5 =
polyfit(WLAN_log_d2more5,WLAN_RSS2more5,1);
WLAN_lineRSSmore5 = polyval(WLAN_coeffRSSmore5,7:18);
WLAN_alpha2 = -1*WLAN_coeffRSSmore5(1)

subplot(2,1,2);
plot(WLAN_log_d2less5,WLAN_RSS2less5,'o',0:7,WLAN_lineRSSle
ss5,'-');
hold on
plot(WLAN_log_d2more5,WLAN_RSS2more5,'o',7:18,WLAN_lineRSSm
ore5,'-');
WLAN_title = sprintf('%s\n%s%2.2f%s%1.2f%s%1.2f','WLAN
Calculation Best Fit Line', ...
```

```
    'Lo = ',WLAN_L0 + 8,'   alpha1 = ',WLAN_alpha1,'
alpha2 = ',WLAN_alpha2);
title(WLAN_title);
xlabel('10log10(distance)');
ylabel('RSS(dBm)');
```

B.6 MATLAB Code for Converting the Ray Tracing Generated .RAW file into a .TAP file Compatible with Propsim

```
% File: raw2tap.m
% This matlab code is used to generate a .TAP file from a
.RAW file
% generated by the Ray Tracing Software.  The .TAP file is
used by the
% Propsim to define the impulse respnse of the channel.
%
% Author: Mohammad Heidari, Muhammad Ali Assad

% Complex coefficient
jay = sqrt(-1);

%###########################################################
% I/O to enter the values from the .raw file
% filename = 'prefix_1.raw';
% Alternative to a fixed file name, uncomment the text
below:
filename = input('Enter the filename => ','s');

%------ enter the whole file in one string -----------------
fid = fopen(filename, 'r');
s1 = fscanf(fid, '%c');
fclose(fid);
%------ remove the header characters -----------------------
k = 1;
while s1(k) ~= '_'
    k = k + 1;
end;
```

```
while s1(k) == '_'
    k = k + 1;
end;

j = 0;
for i=k:length(s1)
    j = j + 1;
    s2(j) = s1(i);
end;
%------ convert to numbers and sort -----------------------
s3 = str2num(s2);
rt = sortrows(s3,1);

tk  = rt(:,1);              % time values in seconds
ak  = rt(:,2);              % amplitude values (linear/no
units)
phk = rt(:,3);              % phase values (in radians)
dep_angle = rt(:,4);        % departure angle values (in
radians)
arr_angle = rt(:,5);        % arrival angle values (in
radians)
n_cross   = rt(:,6);        % number of crossings
n_ref     = rt(:,7);        % number of reflections

%##########################################################

npaths = length(tk);                % total number of taps
ck     = ak .* exp(-jay * phk);     % construct the complex
taps
ck_db  = 20 * log10(abs(ck));       % tap amplitude values
in dB
% the maximum value of the amplitudes to add to all taps
maxck_db = max(ck_db) ;
```

```
% normalizing the taps so the max amplitude is always zero
ck_db_new = ck_db - maxck_db ;

% --------------------- Purely for making nice plots ------
---------------
tmin = 10*(floor(1e9*tk(1)/10)-1)*1e-9;
tmax = 10*ceil(1e9*tk(npaths)/10)*1e-9;

min_ckdb = min(ck_db);
min_db1  = 10*floor(min_ckdb/10);
min_db   = max(min_db1, -100);

max_ckdb = max(ck_db);
max_db   = 10*ceil(max_ckdb/10);

% ---- this part is just for test how one can write data
into a file ----

len=length(ak);
for i= 1 : len
    tapnum(i)=i-1;
    tapdel(i)=tk(i);
    tapre(i)=real(ck(i));
    tapim(i)=imag(ck(i));
end

% Since Propsim can only process 24 taps, sorting and
picking the highest 24
% taps for the channel model (edited by: Muhammad Assad)
taparray = [tapdel' ck_db_new];
taparraysort = sortrows(taparray,-2);
tapdelsort = taparraysort(:,1);
tapampsort = taparraysort(:,2);
```

```
% Writing to the file
filename_without_dot = filename(1:strfind(filename,'.')-1);
fid2 = fopen([filename_without_dot,'_answer.tap'],'w');
fprintf(fid2,'; PROPSim Channel Model File, version 1.1
\r\n ');
fprintf(fid2,' \r\n' );
fprintf(fid2,'[Model] \r\n');
fprintf(fid2,'CenterFrequency = 2412000000 Hz \r\n');
fprintf(fid2,'CirUpdateRate = 1029.83 Hz \r\n');
fprintf(fid2,'SampleDensity = 64 \r\n');
fprintf(fid2,'CirCount = 100000 \r\n');
fprintf(fid2,'DistributionSeed = 18467 \r\n');
fprintf(fid2,'Optimise = maximise delay accuracy \r\n');
fprintf(fid2,'ChannelCount = 1 \r\n');
fprintf(fid2,'Continuous = true \r\n');
for l=1:len % since propsim can only have 24 taps
    fprintf(fid2,' \r\n' );
    fprintf(fid2,'[Tap %g] \r\n',l - 1);
    fprintf(fid2,'Description =  \r\n');
    fprintf(fid2,'Delay = constant, %e
\r\n',tapdelsort(l));
    fprintf(fid2,'MeanAmplitude = constant, %e
\r\n',tapampsort(l));
    fprintf(fid2,'StandardModel = classical \r\n');
    fprintf(fid2,'AmplitudeDistribution = rayleigh \r\n');
    fprintf(fid2,'DopplerSpectrum = jakes \r\n');
    if l==24
        break
    end
end
fclose(fid2);
```

References

[1] K. Pahlavan, “Localization interface using WiFi,” RFID Academic Convocation, Las Vegas, Nevada: May 2006.

[2] ISO/IEC 24730, “Real-time locating systems (RTLS),” 2006.

[3] Daniel D. Deavours, “UHF EPC tag performance evaluation.” RFID Journal [Online], May 2005. Available: http://www.rfidjournal.net/Alliance_Lab_Report_intro.pdf

[4] A. Hatami and K. Pahlavan, “Comparative statistical analysis of indoor positioning using empirical data and indoor radio channel models,” IEEE CCNC Proceedings, 2006.

[5] M. Kim, J. J. Fielding, and D. Kotz, “Risks of using AP locations discovered through war driving,” Department of Computer Science, Dartmouth College, 2006.

[6] R. Singh, “Indoor Testbed,” Indoor localization based on Wireless LAN and Heterogeneous sensor fusion [Online], University of Genova, Italy: 2006, Available: https://www.isip40.it/personal/reetu.php.

[7] J. Kolu and T. Jamsa, “A verification platform for cellular geolocation systems,” Elektrobit Ltd., Finland: 2002.

[8] T. Jamsa, T. Putanen, and J. Meinila, “Implementation techniques of broadband radio channel simulators,” Elektrobit Ltd., Finland: 2001.

[9] M. Heidari and K. Pahlavan, “Performance evaluation of indoor geolocation systems using propsim hardware and ray tracing software,” IWWAN Conference, 2004.

[10] K. Pahlavan and A. Levesque, Wireless Information Networks, 2nd ed. New York: John Wiley & Sons, 2005.

[11] T. Holt, K. Pahlavan, and J.F. Lee, “A computer graphics package for indoor radio channel simulation using a 2D ray tracing algorithm,” IEEE Conference Proceedings on Local Computer Networks, September 1992.

[12] A. Hatami, “Application of Channel Modeling for Indoor Localization Using TOA and RSS,” PHD dissertation, Worcester Polytechnic Institute, Worcester: May 2006.

[13] Yan Xu, “Report on Ray Tracing Software,” Directed Research under Professor K. Pahlavan, Worcester Polytechnic Institute, Worcester: January 1998.

[14] J. Medbo and J-E. Berg, “Measured radiowave propagation characteristics at 5 GHz for typical HIPERLAN/2 scenarios,” ETSI/BRAN document no. 3ERI084A.

[15] J. Medbo and P. Schramm, “Channel models for HIPERLAN/2,” ETSI/BRAN document no. 3ERIO85B.

[16] V. Erceg, et al., “TGn channel models,” IEEE P802.11 Wireless LANs Document, May 2004.

[17] M. Gudmundson, “Correlation model for shadow fading in mobile radio systems,” Electronics Letters, vol. 27, no. 23, pp. 2145-2146, September 1991.

[18] J. Makela, T. Braysy, and K. Pahlavan, “Analysis of the effects of handoff on the performance of tactical communication systems using WLANs,” IEEE MilCom, 2005.

[19] A. Hatami and K. Pahlavan, “A comparative performance evaluation of RSS-Based positioning algorithms used in WLAN networks,” IEEE WCNC Conference, 2005.

[20] E. D. Kaplan, *Understanding GPS: Principles and Applications*, Artech House, Boston, MA: 1996.

[21] M. Heidari, K. Pahlavan, “Performance Evaluation of WiFi RFID Localization Technologies,” *RFID Technology and Applications,* Cambridge University Press, 2007.

[22] M. Kanaan, M. Heidari, F. O. Akgul, and K. Pahlavan, “Technical aspects of localization in indoor wireless networks,” Bechtel Telecommunications Technical Journal, 2006.

[23] K. Pahlavan, M. Heidari, F. O. Akgul, and A. Hatami, “Indoor geolocation in the absence of direct path,” IEEE Wireless Communications, 13:6, pp. 50-58, 2006.

[24] B. Alavi, “Distance measurement error modeling for time-of-arrival based indoor geolocation,” PHD Disseration, Worcester Polytechnic Institute, Worcester: April 2006.

[25] A. Malekitabar, H. Aghababa, M. H. Radfar, and B. H. Khalaj, “Minimizing the error of time difference of arrival method in mobile networks,” IEEE, 2005.

[26] Wikipedia, “Radio-frequency identification [Online],” Last modified: May 2007, Available: http://en.wikipedia.org/wiki/RFID

[27] Global Information Inc., “Real Time Locating Systems 2006-2016 (RTLS),” ID Tech Ex Ltd., 2006.

[28] B. Bacheldor, “ISO ratifies 2.4 GHz RTLS standard,” RFID Journal, April 2007.

[29] Elektrobit, Propsim C8 Key Features [Online], Available: http://www.propsim.com/index.php?1992.

[30] Ekahau, Ekahau Positioning Engine 4.0 Overview [Online], Available: http://www.ekahau.com/?id=4500.

[31] Ekahau, Ekahau T201 WiFi Tag Datasheet [Online], Available: http://www.ekahau.com/file.php?id=99156.

[32] Ekahau, RTLS Overview [Online], Available: http://www.ekahau.com/?id=4200.