Stony Brook University

Wireless Indoor Localization using Expectation-Maximization on Gaussian Mixture Models

A Thesis Presented

by

Abhishek Goswami

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Master of Science

in

Computer Science

Stony Brook University

May 2011

Stony Brook University

The Graduate School

Abhishek Goswami

We, the thesis committee for the above candidate for the Master of Science degree, hereby recommend acceptance of this thesis

Samir Das - Thesis Advisor
Professor, Dept. of Computer Science

Luis Ortiz
Assistant Professor, Dept. of Computer Science

I.V. Ramakrishnan
Professor, Dept. of Computer Science

This thesis is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Thesis

Wireless Indoor Localization using Expectation-Maximization on Gaussian Mixture Models

by

Abhishek Goswami

Master of Science

in

Computer Science

Stony Brook University

2011

We consider the problem of localizing a wireless client in an indoor environment based on the signal strength of its transmitted packets as received on stationary sniffers or access points.

Current state-of-the art indoor localization techniques have the drawback that they rely extensively on a 'training phase'. This 'training' is a labor intensive process and must be done for each target-area under consideration for various device types. This clearly does not scale for large target areas. The introduction of unmodeled hardware with heterogeneous power-levels etc further reduces the accuracy of these techniques.

We propose a solution in which we model the received signal strength as a Gaussian Mixture Model (GMM). We use expectation maximization to find the parameters of our GMM. We can now give a location fix for a transmitting device based on the maximum likelihood estimate. This way, we not only avoid the costly 'training phase' but also make our location estimates much more robust in the face of various form of heterogeneity and time varying phenomena. We present our results on two different indoor testbeds (CEWIT and Computer Science Buildings in Stony Brook University) with multiple WiFi devices (iphones, android, laptops, netbooks). We demonstrate that the accuracy is at par with state-of-the-art techniques but without requiring any training.

We also show an application of such localization in extracting the hidden social structure of the occupants of the building based on their WiFi activity. We show interesting observations from the Computer Science building in Stony Brook University.

Contents

List of Figures

Acknowledgment

I would like to express my deep sense of gratitude to **Prof. Samir Das** and **Prof. Luis Ortiz**, for their invaluable help and guidance during the course of this work. I am highly indebted to them for constantly encouraging me by giving their critics on my work. I am grateful to them for having given me the support and confidence.

Abhishek Goswami
May 2011
Stony Brook University

Chapter 1

Introduction

Devices with wireless cards e.g Laptops, PDAs etc are increasingly becoming immensely popular. Infact many enterprises and office locations have adopted a 'wire-free' model and provide Wi-Fi access to all employees / occupants of the building. Wireless devices enable mobility for the user, which in turn creates a need for location aware applications. It is possible to extract the signal strength of 802.11 wireless frames being transmitted by a Wi-Fi device (both APs and clients) This has motivated the use of observed signal strength as a parameter for performing localization of wireless devices.

1.1 WLAN Localization Architecture

There are two ways of looking at the localization problem: a client-based approach and a server-based approach. In the client-based model, the client is the active entity. The localization algorithm runs on the client device and localization is typically performed based on the wireless LAN characteristics being seen by the client at that location. The need for the wireless client to download, install and run extra software can be a concern in a power-constrained environment. Client-based localization techniques can be used only when the wireless user is interested in being localized.

In a server-side model, the client is a passive entity. The localization algorithm is executed on a backend server. The server-side techniques typically use other devices in the network (e.g. APs/sniffers etc) to capture packet transmissions made by the client device. Server-side techniques are particularly interesting because they do not require any modification to the hardware or software of the client being tracked. For security and management applications, a server-based approach is more suitable. This approach does however raise questions on location-privacy because a client device may be localized without the user being aware of it.

This thesis presents a server-side technique that gives a location-fix based on the Received Signal Strength (RSS) information obtained from sniffer packet captures.

1.2 Motivation of the Project Idea

We observe here that RF-based systems need to deal with the noisy characteristics of the wireless channel. This has motivated the use of various localization techniques for WLAN-based location sensing. Such techniques usually work in two phases: an *offline* training phase and an *online* location determination phase.

RADAR [10] was one of the first RF-based indoor localization schemes. In RADAR, the authors suggest two deterministic schemes for localization. The first scheme has an offline training phase and uses the nearest neighbor in signal space (NNSS) as the metric to compare the multiple locations on the map and pick the one that best matches the observed signal strength vector. The second scheme does not rely on the offline training phase and instead relies on a mathematical model of indoor signal propagation to generate a set of theoretically-computed signal strength data for each location in the target space. The NNSS metric is then used to estimate the location of the mobile user by matching the observed RSS to the theoretically computed SS at these locations.

Recently, a number of probabilistic techniques have been used for WLAN-based location sensing. In such techniques, the offline phase corresponds to the construction of conditional probability distributions which map signal intensities to locations on a map. Thus, we first build up a *signal map* database for the area being covered. During the location determination phase, given a real-time RSS-signal vector of the target device, we use a probabilistic inference algorithm to select the most likely location from all possible locations in the target area.

There are a number of challenges in existing probabilistic localization techniques. One, there needs to be a trained point for each possible target location on the map. Training requires a lot of time-consuming (usually manual) effort. Moreover, training at each discretized location on the map clearly does not scale if the target area is large. Plus, there may be locations where we may not have direct access to - e.g. an office room with restricted access etc. These points would not be covered during training and would subsequently never show during localization. Two, for final location estimation, probabilistic techniques depend heavily on the data collected during the *offline* training phase. The parameters of the model are calculated from the data collected during the training phase. These parameters are fixed for each trained location. Not having dynamic parameters for the model can substantially reduce the accuracy of the location estimates in the presence of time varying phenomena like movement of people inside the building, other active devices in the vicinity etc. Third, wireless characteristics vary substantially depending on the hardware being used. Using a specific wi-fi card for training effectively binds us to that hardware. This reduces the flexibility and robustness of localizing client devices with unmodelled hardware, devices operating at varying power levels etc. These issues serve as the motivation for this thesis.

In this project, we present a server-side indoor location-sensing system using prob-

abilistic techniques. The idea behind the proposed algorithm is to first initialize the parameters of the model using a naive indoor radio propagation model. We then update the parameters of the model based on data samples collected during a sliding time-window. Thus we use the observed data itself to give us a better estimate of the parameters of our model. We then use these optimized parameters to localize clients observed during the time-window. This way, we not only avoid the costly training phase but also make our location estimates much more robust in the face of time varying phenomena. Also, we have effectively removed the restriction of having a set of specific hardware for training. This makes our algorithm much more generic and we can now use it to localize any device equipped with a Wi-Fi interface.

1.3 Organization Of the Report.

- Chapter 2 discusses related work in the field of indoor Wi-Fi localization.
- Chapter 3 we discuss some interesting characteristics of the wireless channel that we incorporate in our model to solve the localization problem.
- Chapter 4 presents our problem formulation in terms of a Gaussian Mixture Model.
- Chapter 5 presents the EM algorithm from the perspective of our problem formulation.
- Chapter 6 gives details on our testbed and dataset.
- Chapter 7 presents the results of our technique and how they compare with other existing techniques.

Chapter 2

Related Work

Some calibration-free techniques have been proposed [5] [6] [18] etc. The objective of such techniques is to automate the effect of wireless physical characteristics on RSS measurements and make them responsive to environmental dynamics like temperature and humidity variations, furniture variation, human mobility etc. This is usually done by having reference Access Points (or sniffers) deployed in the target space and then measuring RSS between the 802.11 APs and also between a client and its neighbouring APs (or sniffers). In [5] Moares et al use an indoor signal propagation model to generate a *radio propagation map (RPM)* at each sniffer. Thereafter they use RSS measurements between the sniffers and a reference Access Point(AP) to reconstruct the RPM, either periodically or when there are significant variations of RSS values. In [18] Lim et al. use the on-line RSS measurements to create a mapping between the RSS measure and the actual geographical distance.

Such techniques are essentially modelled to capture real-time changes in the environmental dynamics of the target space. But they do not model variations in client hardware and transmission power which can significantly degrade the positional accuracy of RSS based Wi-Fi localization schemes.

In [15] Tsui et al. also observe that hardware variance can significantly degrade the positional accuracy of RSS-based Wi-Fi localization systems. Infact they note that the hardware variance problem is not limited to differences in the WiFi chipsets used by training and tracking devices but also occurs when the same Wi-Fi chipsets are connected to different antenna types and/or packaged in different encapsulation materials. The authors stick to the *online*-training and *offline* location-determination model but add an intermediate online-adjustment phase . In this intermediate phase they use unsupervised learning methods to construct a signal transformation function between the training device and a new tracked device.

In [16] Tao et al. have an interesting take on unmodelled-hardware and transmission power variations being effected by a transmitting client. They also stick to the *online*-training and *offline* location-determination model.However, they observe that RSS is lin-

early proportional to transmission power. Thus the difference in received signal strengths between a pair of sniffer devices would not vary dramatically as the transmission power of a client device changes. Based on the difference in signal strength between every pair of sniffers, they suggest a weighted heuristic to estimate a location-fix for a given target RSS fingerprint. With such a 'difference' based approach, we can no longer assume that the sniffers are independent. Thus, we are restricted to the use of a heuristic in this model. However, the observation that RSS is linearly proportional to transmission power is very interesting. Infact, we use this observation in building our model.

The major contribution of this work is to develop an algorithm that does not rely on training data. Instead, the algorithm can learn the parameters of the model from real-time transmissions being made by a Tx-client. Thus it can adapt to variations in transmit power across heterogeneous devices which makes it particularly suitable for server-side localization techniques. Plus this model can also factor in real-time changes in the environmental dynamics of the target space.

Chapter 3

Wireless Characteristics

Our system is based on the 802.11 wireless networking protocol, which is inexpensive and widely deployed in enterprise offices and academic campuses. 802.11 uses 11 channels in the ISM band. Signal propagation in this band is complex and in this section, we identify the different causes of variation in the wireless channel quality and how we factor them into our model. Our approach is server-based, where we capture client packets using sniffers. As such, we are mainly concerned with the variations that affect the Received Signal Strength (RSS) on the sniffer. In this section, experimentally validate two observations that have been made previously in wireless-localization literature. We model our problem around these two observations.

3.1 Distribution of Signal Strength

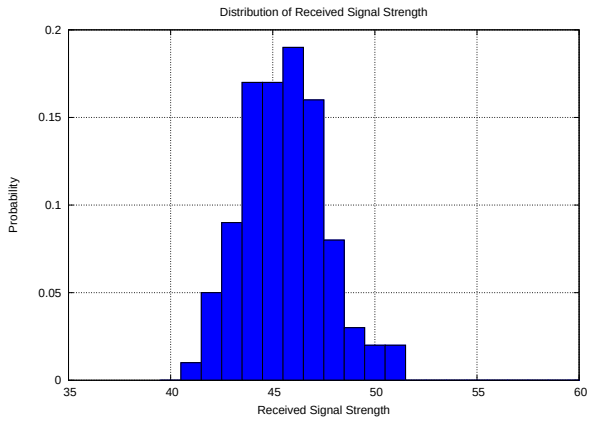


Figure 3.1: Received Signal Strength at a sniffer from a laptop operating at a fixed power-level

Figure 3.1 shows the distribution of Received Signal Strength values observed by a sniffer located a fixed distance apart from a transmitting client. The Tx-client is a Dell laptop having a Ubiquiti XR2 wireless card and is using a fixed power-level for wireless transmissions.

We observe that the Signal Strength distribution is roughly Gaussian. In [16] et al also make similar observations. [8] [5] etc also model signal intensity as a normal distribution.

3.2 Transmission Power

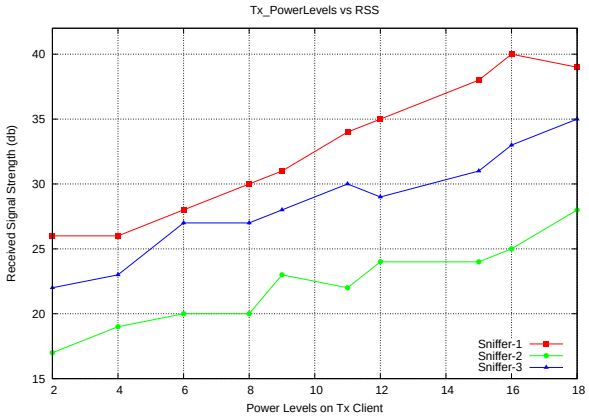


Figure 3.2: Signal strength readings from three different receivers of a signal from a signal transmitter, with the transmitter varying its Tx-Power

Figure 3.2 shows how the observed signal strength changes as the transmission power is varied. Our experiments validate the observations made in [16] by Tao et al in that the observed signal strength is linearly proportional to the transmission power.

Chapter 4

Problem Formulation

The Gaussian Mixture Model is a simple linear superposition of Gaussian components, aimed at providing a richer class of density models than the single Gaussian. We now formulate our problem as a Gaussian mixture in terms of discrete latent variables.

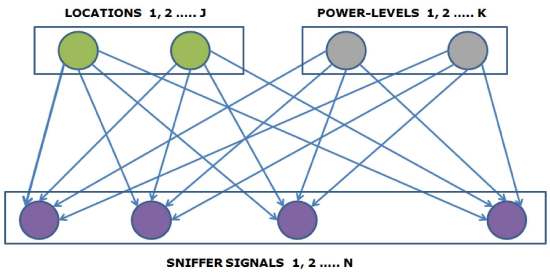


Figure 4.1: Gaussian Mixture Model

4.1 Latent Variables for Target Locations and Power Levels

We introduce a J-dimensional binary random variable $\mathbf{x}$ representing possible target locations. $\mathbf{x}$ has a 1-of-J representation in which a particular element x_j is equal to one and all other elements are equal to 0. The values of x_j therefore satisfy $x_j \in \{0,1\}$ and $\sum_j x_j = 1$. Thus we see that there are J possible states for the vector $\mathbf{x}$

The probability distribution over $\mathbf{x}$ can be specified as a multinomial

$$p(x_j = 1) = \upsilon_j$$

where the parameters $\{\upsilon_j\}$ must satisfy

$$0 \leq \upsilon_j \leq 1 \; and \; \sum_{j=0}^{J} \upsilon_j = 1$$

Similarly, let us introduce a K-dimensional binary random variable $\mathbf{z}$ representing Power Levels. $\mathbf{z}$ has a 1-of-K representation in which a particular element z_k is equal to one and all other elements are equal to 0. The values of z_k therefore satisfy $z_k \in \{0,1\}$ and $\sum_k z_k = 1$. Vector $\mathbf{z}$ has K possible states.

The distribution over $\mathbf{z}$ is specified as a multinomial

$$p(z_k = 1) = \tau_k$$

where the parameters $\{\tau_k\}$ must satisfy

$$0 \leq \tau_k \leq 1 \; and \; \sum_{k=0}^{K} \tau_k = 1$$

4.2 Constructing the distribution over the observed signal strengths

Let $\mathbf{s}$ be the N-dimensional vector representing the signal strengths observed by the N sniffers placed in the area.

Using the chain rule of probability, we can now define the joint distribution $p(\mathbf{s}, \mathbf{x}, \mathbf{z})$ in terms of the distribution $p(\mathbf{x}, \mathbf{z})$ and the conditional distribution $p(\mathbf{s}|\mathbf{x}, \mathbf{z})$, corresponding to the graphical model in Figure 4.1.

$$p(\mathbf{s}, \mathbf{x}, \mathbf{z}) = p(\mathbf{x}, \mathbf{z})p(\mathbf{s}|\mathbf{x}, \mathbf{z}) \tag{4.1}$$

Moreover $\mathbf{x}$ and $\mathbf{z}$ are independent random variables. So we have

$$\begin{aligned} p(\mathbf{s}, \mathbf{x}, \mathbf{z}) &= p(\mathbf{x}, \mathbf{z})p(\mathbf{s}|\mathbf{x}, \mathbf{z}) \\ &= p(\mathbf{x})p(\mathbf{z})p(\mathbf{s}|\mathbf{x}, \mathbf{z}) \end{aligned} \tag{4.2}$$

Equation 4.2 gives us the joint distribution as $p(\mathbf{x})p(\mathbf{z})p(\mathbf{s}|\mathbf{x}, \mathbf{z})$. The marginal distribution of $\mathbf{s}$ is then obtained by summing the joint distribution over all possible states of $\mathbf{x}$ and $\mathbf{z}$ to give the following probabilistic model :

$$p(\mathbf{s}) = \sum_{\mathbf{x}} \sum_{\mathbf{z}} p(\mathbf{x})p(\mathbf{z})p(\mathbf{s}|\mathbf{x}, \mathbf{z}) \tag{4.3}$$

4.2.1 Independence of Sniffers

We assume the sniffers are independent. This assumption is justified in our model because our sniffers are passive nodes responsible for capturing wireless packets. They have no interaction with each other.

Thus, the term $p(\mathbf{s}|\mathbf{x},\mathbf{z})$ in equation 4.3 can be simplified as

$$p(\mathbf{s}|\mathbf{x},\mathbf{z}) = \prod_{i=1}^{N} p(s_i|\mathbf{x},\mathbf{z}) \tag{4.4}$$

Moreover, from the observations made about Signal Strength variations in Section 3.1 above, the distribution of signal strength can be modelled as a Gaussian determined by the (location, power-level) pair.

That is

$$s_i|(x_j, z_k) \sim gaussian(\mu_{i\ (j,k)}\ , \sigma_{i\ (j,k)})$$

This lends simplicity to our model since the term $p(\mathbf{s}|\mathbf{x},\mathbf{z})$ in equation 4.4 can be further simplified as

$$p(\mathbf{s}|\mathbf{x},\mathbf{z}) = \sum_{j=1}^{J}\sum_{k=1}^{K}(\prod_{i=1}^{N}\mathcal{N}[s_i|\mu_{i\ (j,k)}\ , \sigma_{i\ (j,k)}]) \tag{4.5}$$

4.3 Model Parameters

Putting equation 4.3 and equation 4.5 together we get the distribution of $\mathbf{s}$ as

$$p(\mathbf{s}) = \sum_{j=1}^{J}\sum_{k=1}^{K}(\upsilon_j\tau_k\prod_{i=1}^{N}\mathcal{N}[s_i|\mu_{i\ (j,k)}\ , \sigma_{i\ (j,k)}]) \tag{4.6}$$

Thus we have modelled the marginal distribution of $\mathbf{s}$ as a Gaussian mixture with target locations and power levels as our latent variables. The parameters of our model are

$$\theta = \big(\upsilon_j, \tau_k, (\mu_{i\ (j,k)}, \sigma_{i\ (j,k)})\big)$$

where $j \in \{1, ...J\}$, $k \in \{1, ...K\}$ and $i \in \{1, ...N\}$. We now use the Expectation Maximization(EM) algorithm to estimate the parameters of our model.

Chapter 5

EM Algorithm

An elegant and powerful method for finding maximum likelihood solutions for models with latent variables is the Expectation Maximization(EM) algorithm. The EM algorithm is an iterative process through two steps: an expectation step(E-step) and a maximization step(M-step). During the iterations, a sequence of model parameters $\theta^{\mathbf{0}}$, $\theta^{\mathbf{1}}$, ..., θ^{*} is generated where $\theta^{\mathbf{0}}$ is the initial parameter and θ^{*} is the converged parameter obtained when the algorithm terminates.

5.1 E-step

Suppose we have a data set of observations $\overline{\mathbf{S}}$ = { $\mathbf{s}^0$, $\mathbf{s}^1$, ...,$\mathbf{s}^M$}. The E-step corresponds to finding the expected value of the hidden component ($\mathbf{x}$ and $\mathbf{z}$) values given the observed data $\overline{\mathbf{S}}$ and the current parameter estimates.

Using this observation set and the current parameter estimates, we find out the posterior probabilities (or responsibilities) as follows.

For each observation $\mathbf{s}^l$

$$\pi^l_{(x_j,z_k)} \equiv p(x_j = 1, z_k = 1|\mathbf{s}^l) \tag{5.1}$$

$$= \frac{p(x_j = 1)p(z_k = 1)p(\mathbf{s}^l|x_j = 1, z_k = 1)}{\sum_{p=1}^{J} \sum_{q=1}^{K} p(x_p = 1)p(z_q = 1)p(\mathbf{s}^l|x_p = 1, z_q = 1)} \tag{5.2}$$

$$= \frac{\upsilon_j \ \tau_k N(\mathbf{s}^l|\mu_{j,k}, \sigma_{j,k})}{\sum_{p=1}^{J} \sum_{q=1}^{K} [\upsilon_p \tau_q N(\mathbf{s}^l|\mu_{p,q}, \sigma_{p,q})]} \tag{5.3}$$

The posterior probability value $\pi^l_{(x_j,z_k)}$ can be viewed as the *responsibility* that component (x_j, z_k) takes for explaining observation $\mathbf{s}^l$. We find out this measure of responsibility for each observation in our data set $\overline{\mathbf{S}}$.

5.2 M-step

The M-step of the algorithm corresponds to maximizing the likelihood of the observed data. This leads us to re-estimating the parameters for the next iteration based on the posterior probabilities calculated in the expectation step of the algorithm.

$$\upsilon_j = \frac{\sum_{l=1}^{M} \sum_k \pi^l_{(x_j,z_k)}}{M}$$

$$\tau_k = \frac{\sum_{l=1}^{M} \sum_j \pi^l_{(x_j,z_k)}}{M}$$

$$\mu_{i\ (j,k)} = \frac{\sum_{l=1}^{M} \pi^l_{(x_j,z_k)} s^l_i}{N_{j,k}}$$

where we have defined

$$N_{j,k} = \sum_{l=1}^{M} \pi^l_{(x_j,z_k)}$$

The variance parameter can also be updated accordingly.

5.3 Convergence of Log Likelihood

Each update of the parameters resulting from an E-step followed by an M-step is guaranteed to increase the log likelihood function. The algorithm is deemed to have converged when the change in the log likelihood function falls below a threshold.

$$\ln p(\overline{\mathbf{S}}|\theta) = \sum_{l=1}^{M} \ln \left\{ \sum_{j=1}^{J} \sum_{k=1}^{K} \upsilon_j \tau_k \mathcal{N}(\mathbf{s}^l|\mu_{j,k}, \sigma_{j,k}) \right\} \tag{5.4}$$

5.4 Handling Identifiability in our Model

In [21] Bishop et al discuss the problem of *identifiability* associated with assigning P sets of parameters to P components. The problem occurs because there are P! ways of assigning P sets of parameters to P components.

In our case each component can be represented as a (location, power-level) pair. We handle the problem of identifiability as follows :

5.4.1 Indoor Radio Propagation Model

The indoor radio propagation model is represented as

$$P_{Rx} = P_0 - 10n \log \left(\frac{d}{d_0} \right)$$

where P_0 is the received signal strength at a distance d_0 from the emitter. P_{Rx} is the signal strength(s_i) seen by receiver for a transmitter located at a distance d away from it. n is a parameter which models the behaviour of the environment. This formula effectively initializes the components representing different locations on the map.

To initialize k components (say) which have a common location but vary in power-level, we make use of the observations made in Section 3.2 which show that the observed signal strength is linearly proportional to the transmission power. Thus, once the formula above gives us the signal value for a specific location, we extrapolate the value linearly to initialize each of the k components for that location

In our experiments, we set n = 2. The corresponding signal strength was used to initialize the means ($\mu_{j,k}$). The standard deviation ($\sigma_{j,k}$) was initialized to 5 (and kept fixed to reduce computation time). As subsequent results show, a value of k = 45 is sufficient to hit a constant average error distance.

5.5 Final Location Estimate

Given a real-time received signal vector $\mathbf{s}^{(obs)}$, we can now find the location with the highest probability. We do this by first finding the probability for each (location, power-level) pair and then marginalizing over the power-levels. Thus the estimated location index is given by j^* where

$$j^* = max_j \sum_k P(x_j = 1, z_k = 1 | \mathbf{s}^{(obs)})$$

Chapter 6

Testbed and DataSet Details

We begin with a description of our system architecture. We then describe the two testbeds where we conducted our experiments. This is followed by a overview of the components of our sniffer devices. We round up this section by discussing the data collection process.

6.1 System Architecture

As mentioned briefly in Section 1.1 our work is based on a server-side architecture for WLAN localization. Our system consists of a centralized server and a number of sniffer devices. Sniffers provide overlapping coverage for the target area (similar to how APs are usually deployed inside buildings). The sniffers are used to capture packet transmissions made by the client device. The sniffers are connected to a backend server using a power-line ethernet LAN. The sniffer packet captures are transmitted to the server. For each packet, the server records the mac-address of the Tx-client and the corresponding signal strength for that packet.

In the future, our sniffer functionality might be integrated directly into the WLAN APs of a production network. Enterprise APs usually have a centralized controller which can serve as our localization engine. This makes our architecture particularly interesting. Moreover, server-side architectures have the added advantage of allowing us to localize a client device independent of any hardware or software on the laptop.

6.2 Sniffer Information

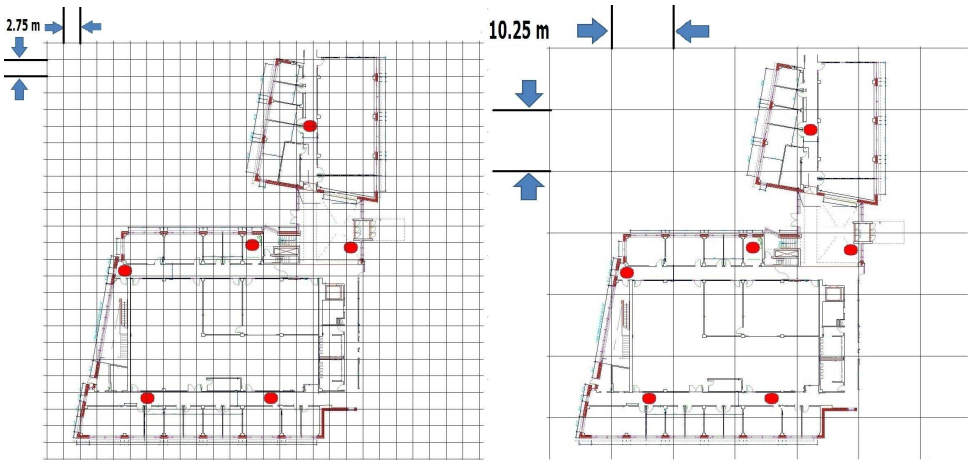
Our sniffer devices are responsible for capturing wireless transmissions made by a Tx-client. We use soekris-net4801 boards as our sniffer devices with atheros-based cm9 cards for wireless captures. Our sniffers are running Pyramid Linux (version 2.6.16-metrix) and we use the default MadWiFi driver which comes with this distribution (0.9.4.5 : svn 1485).

To capture packets we use the Tcpdump software (version 4.0.0 libpcap version 0.9.8)

To obtain signal strength information, the MadWiFi driver allows a monitor mode interface to be created and configured with RadioTap header support. From the radio-tap header we can extract the Received Signal strength of each packet received by the sniffer. We verified that the MadWifi driver had a fixed noise-floor in each of our cm9 cards (-95 dbm). In fact the received signal strength of a frame reported by the MadWiFi driver is actually the SNR value (in db) obtained after subtracting the noise-floor from the raw signal strength value. We work directly with the RSSI value (in db) as reported by the driver.

6.3 Experimental Testbed

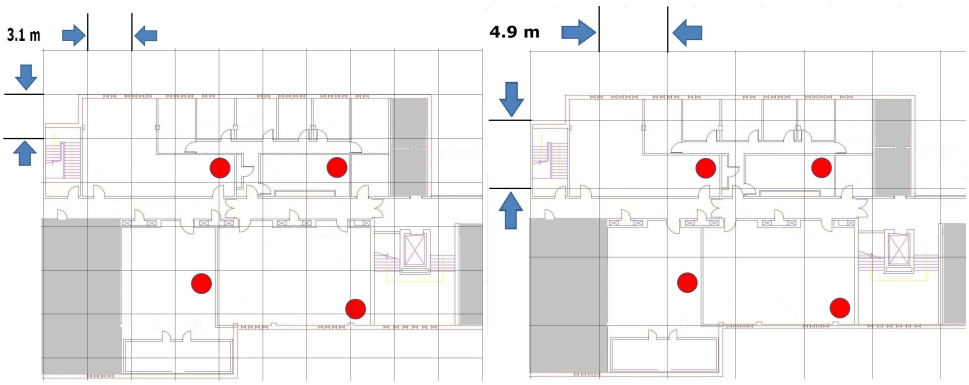
We use two different testbeds to Experimentally validate our technique. Figure 6.1 shows the CEWIT building with a dimension of 50 x 65 meter square. Figure 6.2 shows the CS department building at SBU with dimensions 20 x 30 meter square. The red-dot denotes the position of the sniffers. The sniffers are connected to a backend server using a power-line ethernet LAN.



(a) Fine-grained discretization (b) Coarse-grained discretization

Figure 6.1: Map of the CEWIT Building where experiments were conducted

Figure 6.1(a) and 6.2(a) shows the discretized target space that we use in our algorithm. Our technique does not use training, which allows fine grained discretization of the target space. Figure 6.1(b) and 6.2(b) shows the corresponding discretized space that we use when we present our results in 7.4 when we compare our technique with other techniques that use training data from specific locations of the target space. The coarse granularity of 6.1(b) and 6.2(b) serves to highlight the fact that training is a huge bottleneck in such 'training-based' models. It is usually a manual effort where we have to go to each discrete location on the map and transmit a specified number of packets from that location to construct the *signal map* of the area being covered. Thus we invariably have to resort to coarse discretization when the target space is large.



(a) Fine-grained discretization (b) Coarse-grained discretization

Figure 6.2: Map of the CS Dept Building where experiments were conducted

6.4 Data Collection

We perform our experiments with 4 different wireless clients - an android phone, an iphone, a dell laptop and a dell netbook. We select 50 locations from the CEWIT testbed (Fig 6.1) and 30 locations from the CS Dept testbed (Fig 6.2) and transmit 200 ping packets from each location for each of the above four devices. Each ping carries a sequence number and the pings are spaced apart uniformly, at a rate of 1 per second. The sequence number is used to form the vector of RSS values from each transmission. Thus for each location on the map and for each device, we have a set of 200 RSS tuples to experimentally validate our algorithm.

Chapter 7

Evaluation

We implement our algorithm based on the EM algorithm and collect accuracy estimates on the data sets collected from both testbeds.

We generate plots for the following experiments:

1. **Number of power-levels used for EM:**
   This experiment serves to give us the value of k (the number of power-levels) that we should use in our algorithm.

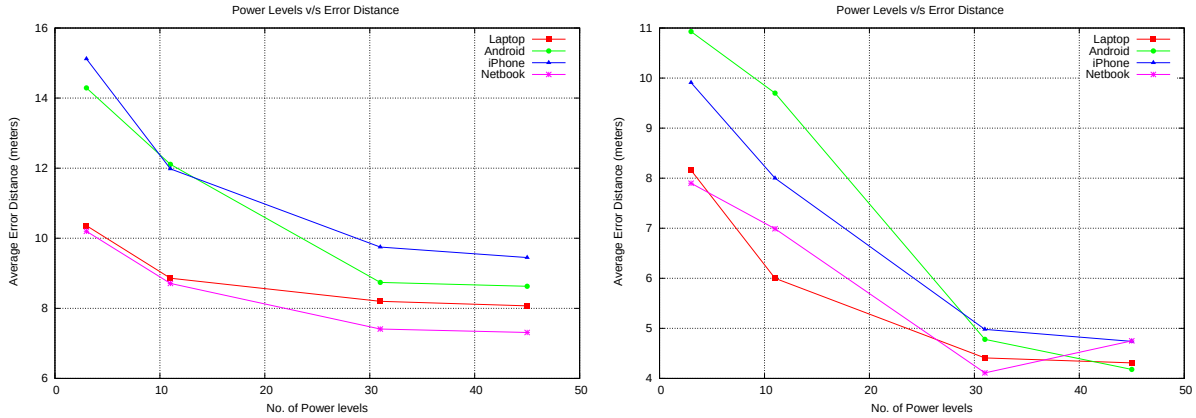
2. **Size of the learning set:**
   This experiment shows how the average error distance varies as a function of the learning set size.

3. **Baseline Comparisons:**
   This set of experiments is used to compare our technique with a baseline Model-based scheme, both of which be applied on the fine-grained discretized target space.

4. **Comparisons with schemes that build RF signal maps:**
   Here we compare our technique with two schemes that use an *offline phase* to first build an RF signal map of the target space.

5. **Mobility-related experiments:**
   These experiments show how the mobility of the Tx-client effects the accuracy of our algorithm.

7.1 Avg. error distance v/s Number of power-levels used for EM



(a) CEWIT (b) CSD

Figure 7.1: Number of power levels v/s Avg Error distance

We see that the avg. error distance does not vary much after we use k=45 in our EM algorithm. The subsequent plots shown here have been generated using k = 45 in the algorithm.

7.2 Learning Set Size

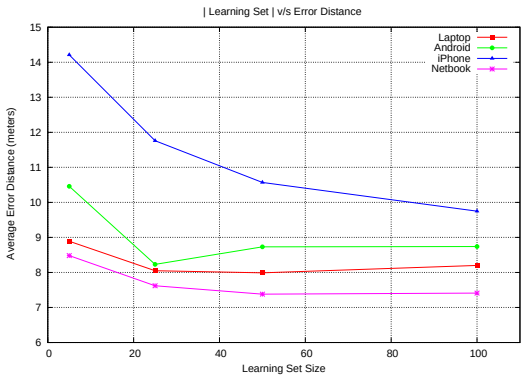


Figure 7.2: Learning Set Size v/s Error distance on CEWIT Dataset

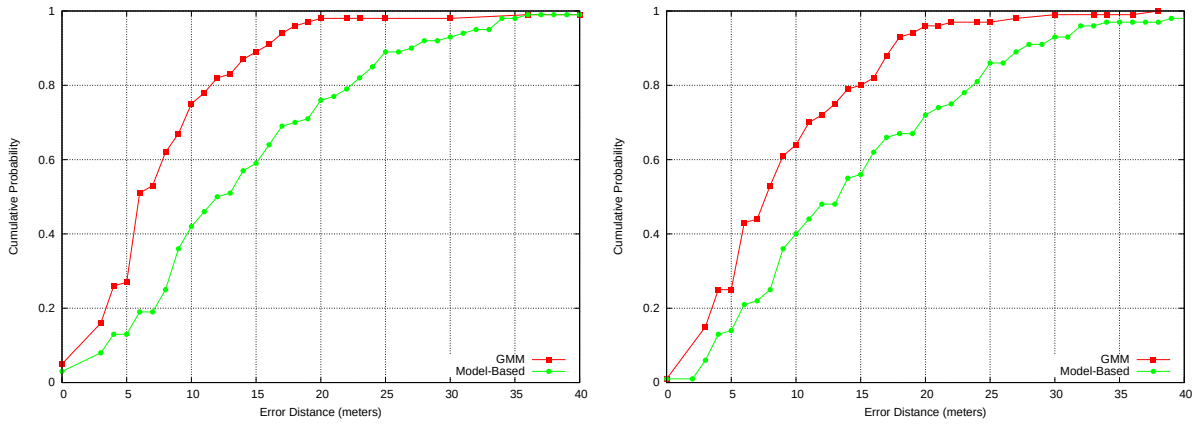
As mentioned in Section 6.4 for every device, we have 200 RSS vectors for each location on the map. Figure 7.2 shows how the Average Error varies for different sizes of the learning set i.e if we use m samples to learn the parameters of the model which we subsequently use to localize the remaining (200 - m) samples for each location. We see that the average error reaches almost hits a plateau after 50 learning samples. The subsequent plots shown here have been generated using 50 RSS samples to learn the model

parameters. We then use these parameters to localize the remaining 150 samples for each location.

7.3 Baseline Comparison

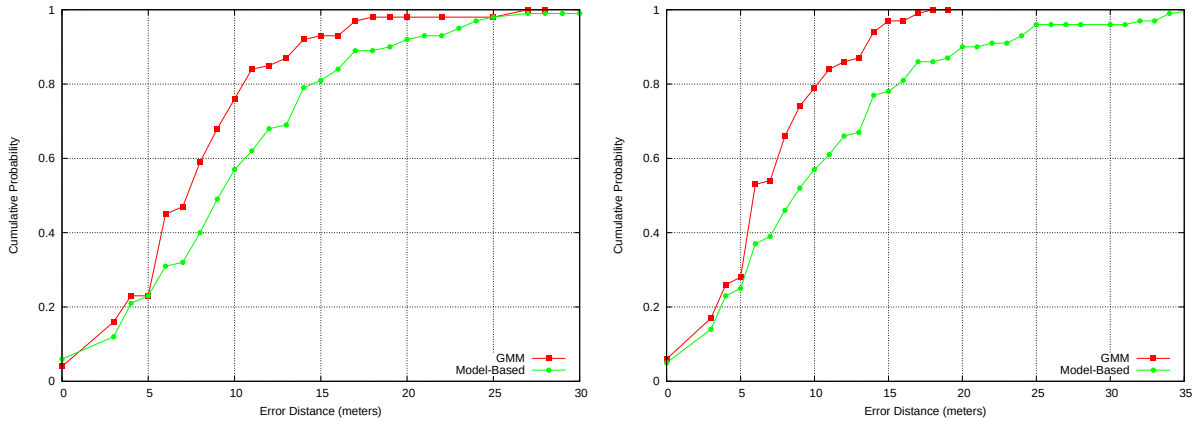
Here we compare our technique with a baseline Model-based technique that can be applied on the fine-grained discretized target space shown in fig 6.1 (a) and 6.2 (a) . The log-distance path loss (LDPL) mentioned in Section is used to predict the RSS at each square vertex that lies inside the target-space. The baseline uses directly uses these values with NNSS as the metric to compare the multiple locations on the map and pick the one that best matches the observed signal strength vector. Our algorithm instead uses these values to initialize our algorithm. For each device, our algorithm uses 50 RSS samples from each location to update the parameters of our model. The new parameters are then used to localize the remaining 150 RSS samples from each location.

7.3.1 CEWIT-Dataset



(a) Android (b) Iphone

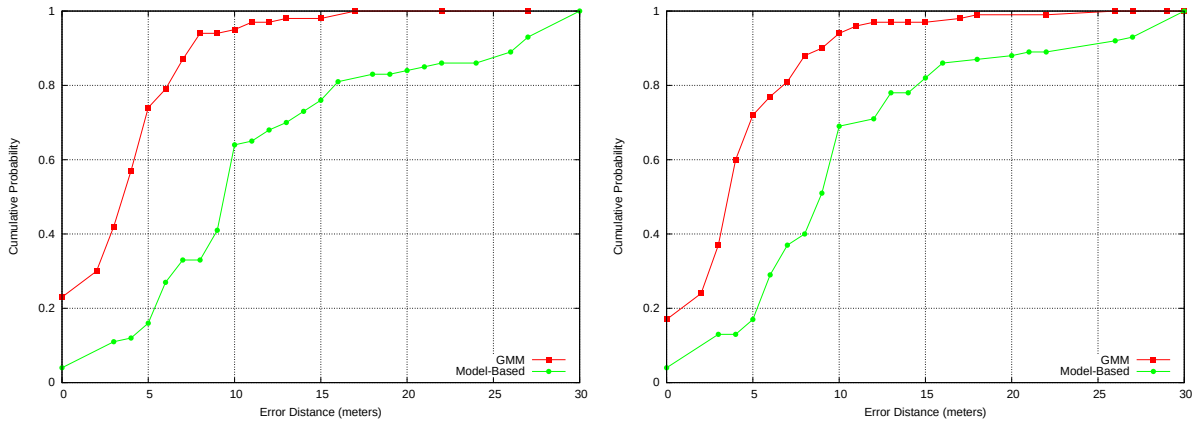
Figure 7.3: Baseline Comparison - Android , Iphone



(a) Dell Laptop (b) Dell netbook

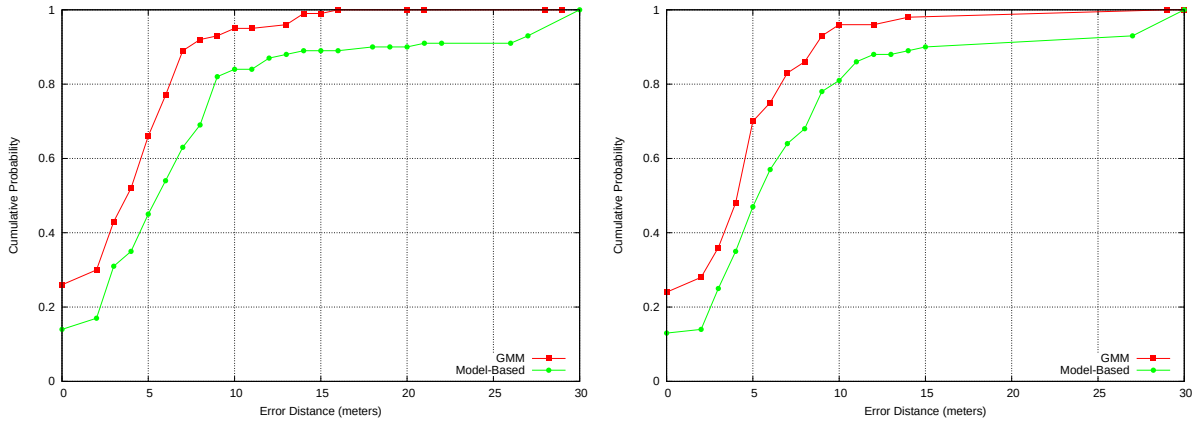
Figure 7.4: Baseline Comparison - Dell Laptop , Dell netbook

7.3.2 CSD-Dataset



(a) Android (b) Iphone

Figure 7.5: Baseline Comparison - Android , Iphone



(a) Dell Laptop (b) Dell netbook

Figure 7.6: Baseline Comparison - Dell Laptop , Dell netbook

We clearly see that we score over the baseline algorithm for all four devices across both the testbeds.

7.4 Comparisons with schemes that build RF signal maps

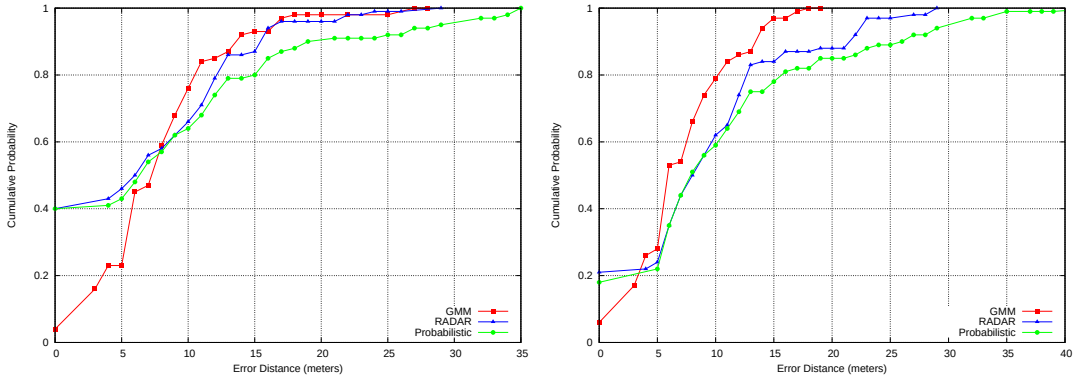
We next compare our technique with two schemes that need to have a training phase to build an RF signal map first. Section explains why we use a coarser granularity as shown in Figure 6.1 (b) and 6.2 (b) to build our signal map.

One of our comparison schemes is deterministic and is based on RADAR [10]. We use NNSS as the metric to identify the location which best matches the observed signal strength vector. The other is a probabilistic scheme on the lines of [8]. Given a location and a sniffer, signal intensity is modelled as a Gaussian distribution based on the training data. We then use a MLE approach to give a location fix for a target RSS fingerprint.

7.4.1 CEWIT-Dataset

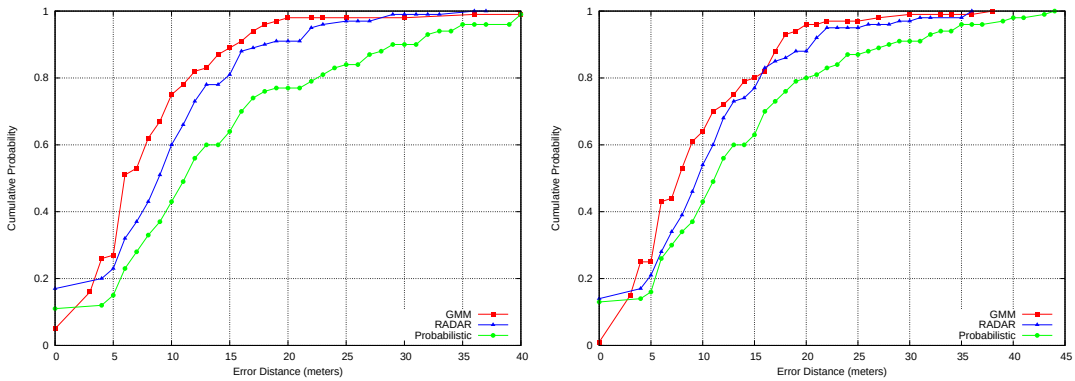
Trainer-Dell Laptop

Test-Dell Laptop, Dell Netbook, Android, Iphone



(a) Dell Laptop (b) Dell netbook

Figure 7.7: Comparison - Dell Laptop , Dell netbook



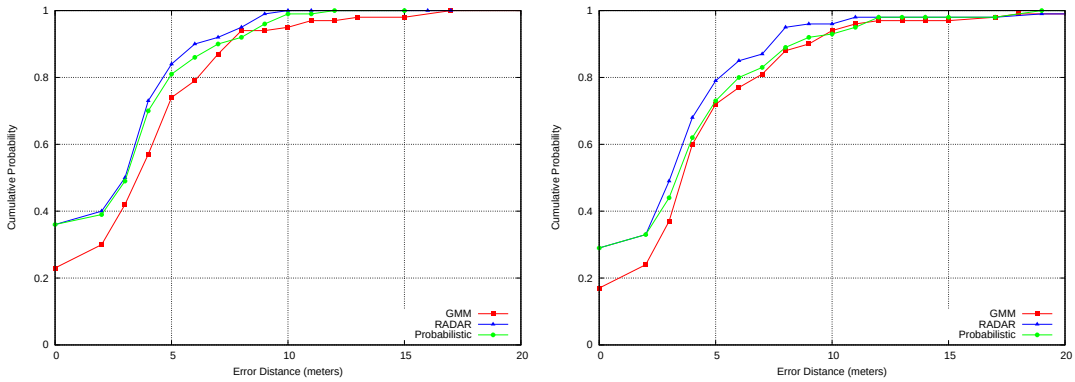
(a) Android (b) Iphone

Figure 7.8: Comparison - Android , Iphone

7.4.2 CSD-Dataset

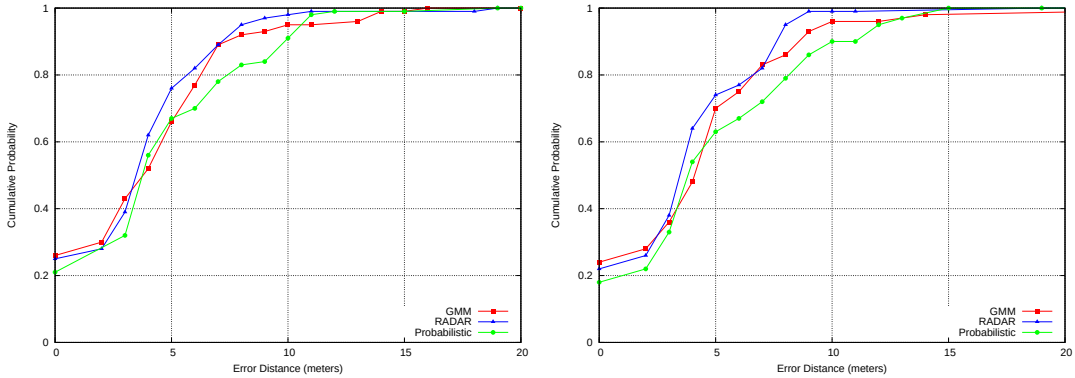
Trainer-Android

Test-Android, Iphone, Laptop, Netbook



(a) Android (b) Iphone

Figure 7.9: Comparison - Android , Iphone



(a) Dell Laptop (b) Dell netbook

Figure 7.10: Comparison - Dell Laptop , Dell netbook

We see that our algorithm performs at-par with state-of-the-art RF-signal-map based techniques especially in cased where the target device is different from the trained device. That is what makes our technique particularly suitable for server-based localization where we do not know the device type of the client being localized.

7.5 Mobility

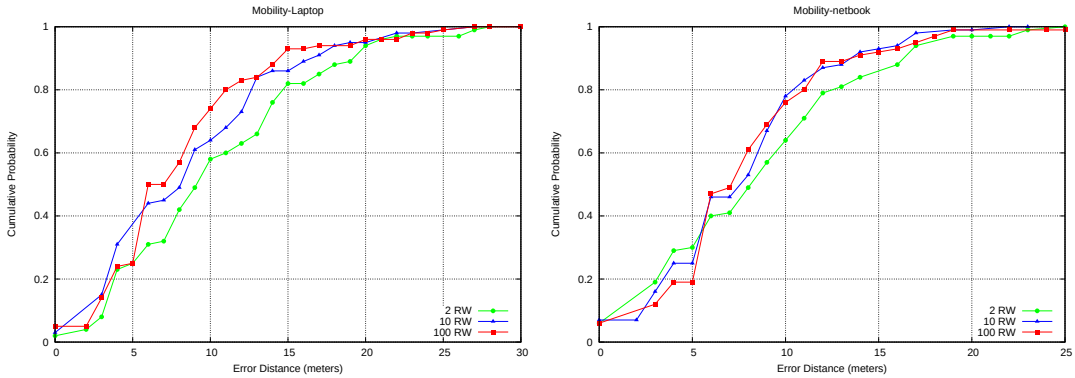
This section shows how the mobility of a client can effect the location estimates made by our algorithm.

In our first experiment, the client makes 2 random walks through the building (i.e a random walk through 50 locations on the CEWIT testbed / 30 locations on the CS Dept testbed) transmitting just a single packet from each location.

In the second experiment, the client makes 10 random walk through the building again transmitting just a single packet from each location.

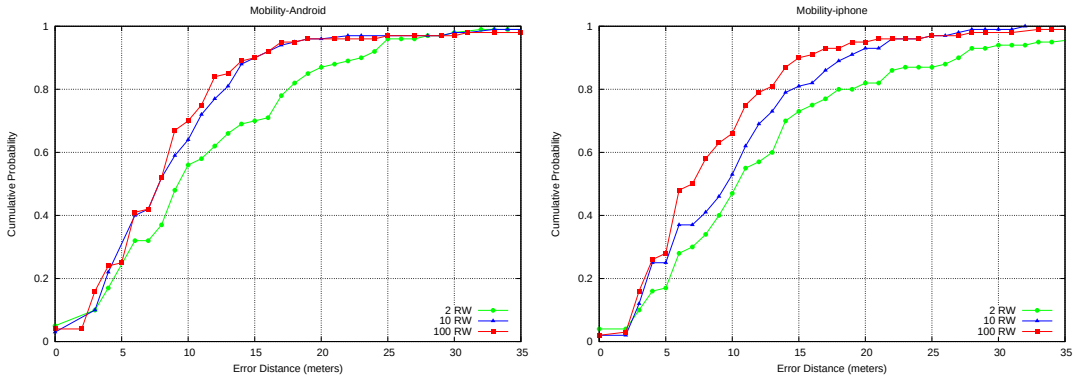
In the last experiment, the client makes 50 random walk through the building again transmitting just a single packet from each location.

7.5.1 CEWIT-Dataset



(a) Dell Laptop (b) Dell netbook

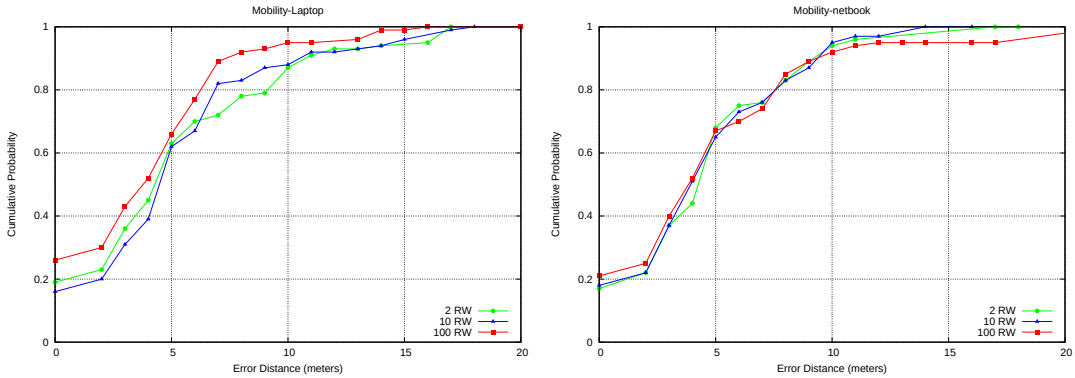
Figure 7.11: Mobility - Dell Laptop , Dell netbook



(a) Android (b) Iphone

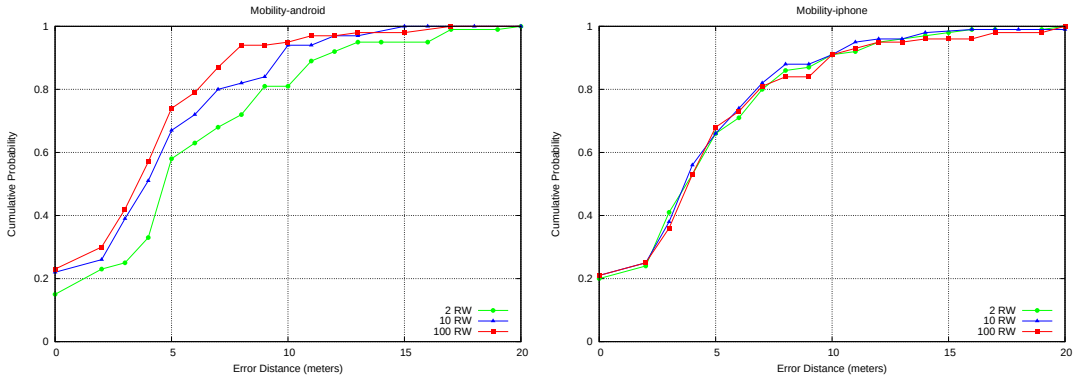
Figure 7.12: Mobility - Android , Iphone

7.5.2 CSD-Dataset



(a) Dell Laptop (b) Dell netbook

Figure 7.13: Mobility - Dell Laptop , Dell netbook



(a) Android (b) Iphone

Figure 7.14: Mobility - Android , Iphone

We see that mobility progressively improves the localization accuracy of our technique. More importantly, in 10 random walks itself we get pretty close to the accuracy estimates obtained from 50 random walks.

Chapter 8

Conclusion

In this work, we have developed a server-side technique to localize a wireless client in an indoor environment based on the signal strength parameter of its transmitted packets. We developed a learning-based algorithm that can learn the parameters of the model dynamically from packets captured by the stationary sniffers / APs inside the building. By using dynamic packet captures for parameter estimation, we can provide location estimates which are much more robust in the face of time varying phenomena like movement of people inside the building, opening closing of doors etc. Moreover, this technique can be used on a host of heterogeneous devices operating at different power levels.

We do not have an explicit *training* phase in our technique. Infact, we showed that we can achieve accuracy that is at par with state-of-the-art techniques that use training to build RF-signal maps first. Thus, our technique not only eliminates the intensive time-consuming (often manual) training phase but also makes our technique scalable for large target spaces.

References

[1] M. Berna et al., "A learning algorithm for localizing people based on wireless signal strength that uses labeled and unlabeled data." *International Joint Conference on Artificial Intelligence, vol. 18, (2003): 1427-1428.*

[2] T. Roos et al., "A probabilistic approach to WLAN user location estimation." *International Journal of Wireless Information Networks 9, no. 3 (2002): 155-164.*

[3] P. Krishnan et al., "A System for LEASE: Location Estimation Assisted by Stationery Emitters for Indoor RF Wireless Networks." *IEEE Infocom 2 (2004): 1001-1011.*

[4] D. Madigan et al., "Bayesian indoor positioning systems." *IEEE Infocom, 2 (2005): 1217-1227.*

[5] L.F.M de Moraes and B. A.A Nunes, "Calibration-free WLAN location system based on dynamic mapping of signal strength." *Proceedings of the 4th ACM international workshop on Mobility management and wireless access, (2006)*

[6] Y. Gwon and R. Jain, "Error characteristics and calibration-free techniques for wireless LAN-based location estimation." *Proceedings of the second international workshop on Mobility management & wireless access protocols, (2004)*

[7] K. Chintalapudi, et al. , "Indoor localization without the pain." *Proceedings of the sixteenth annual international conference on Mobile computing and networking. (2010): 173-184*

[8] Andreas Haeberlen and Algis Rudys, "Practical robust localization over large-scale 802.11 wireless networks." *Proceedings of the sixteenth annual international conference on Mobile computing and networking. (2004): 70-84*

[9] Ming-hua Zhang, Shen-sheng Zhang and Jian Cao, "Probability-based clustering and its application to WLAN location estimation." *Journal of Shanghai Jiaotong University (Science) 13, no. 5 (2008): 547-552.*

[10] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," *IEEE Infocom, vol. 2, (2000): 775-784.*

[11] D. Molkdar, "Review on radio propagation into and within buildings," *Microwaves, Antennas and Propagation, IEE Proceedings H 138, no. 1 (1991): 61-73.*

[12] A. M Ladd et al., "Robotics-based location sensing using wireless ethernet," *Wireless Networks 11, no. 1 (2005): 189-204.*

[13] M. Youssef and A. Agrawala, "The Horus location determination system," *Wireless Networks 14, no. 3 (2008): 357-374.*

[14] E. Elnahrawy, X. Li, and R. P Martin, "The limits of localization using signal strength: A comparative study," *Sensor and Ad Hoc Communications and Networks, (2004): 406-414*

[15] Arvin Tsui, Yu-Hsiang Chuang, and Hao-Hua Chu, "Unsupervised Learning for Solving RSS Hardware Variance Problem in WiFi Localization," *Mobile Networks and Applications 14, no. 5 (October 1, 2009): 677-691.*

[16] Ping Tao et al., "Wireless LAN location-sensing for security applications," *Proceedings of the 2nd ACM workshop on Wireless security (2003): 11-20*

[17] M. A Youssef, A. Agrawala, and A. Udaya Shankar, "WLAN location determination via clustering and probability distributions," *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, (2003): 143-150.*

[18] H. Lim et al., "Zero-configuration, robust indoor localization: Theory and experimentation," *Proceedings of IEEE Infocom, (2006): 123-125.*

[19] JA Bilmesn, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," *International Computer Science Institute, (1998)*

[20] S Borman, "The expectation maximization algorithm - a short tutorial," *http://www.isi.edu/natural-language/teaching/cs562/2009/readings/B06.pdf.*

[21] Bishop, ChristopherM. 2006. *Pattern Recognition and Machine Learning. Springer.*

[22] Brian Ferris, Dieter Fox, and Neil Lawrence, "WiFi SLAM Using Gaussian Process Latent Variable Models," *IJCAI (2007)*

[23] D. Reynolds, Gaussian Mixture Models (MIT Lincoln Laboratory)