Calibration-Free WLAN Location System Based on Dynamic Mapping of Signal Strength

Luís Felipe M. de Moraes
Federal University of Rio de Janeiro - UFRJ
C.P. 68.511, CEP: 21945-970
Rio de Janeiro-RJ, Brazil.
moraes@ravel.ufrj.br

Bruno Astuto A. Nunes
Federal University of Rio de Janeiro - UFRJ
C.P. 68.511, CEP: 21945-970
Rio de Janeiro-RJ, Brazil.
bruno@ravel.ufrj.br *

ABSTRACT

In this work we present a *calibration-free* system for locating wireless local area network devices, based on the radio frequency characteristics of such networks. Calibration procedures are applied in a great number of proposed location techniques and are considered to be not practical or a considerable barrier to wider adoption of such methods. Thus, we addressed issues related to some aspects of location systems through, an architecture based on wireless sniffers and by constructing a location model based on signal propagation models, in which its parameters are calculated in real time. This guarantee good self-sufficiency and adaptation capacity to the proposed system, once it does not need human intervention to work, neither from the network administrator or the wireless user being located. Moreover, a probabilistic method was used for estimating wireless devices positions, based on the previous constructed model. We later demonstrate the feasibility of our approach by reporting results of field tests in which the proposed technique was implemented and validated in a real-world indoor environment.

Categories and Subject Descriptors

C.2.1 [**Computer-communication Networks**]: Network Architecture and Design—*Wireless communication, Network topology*; C.2.3 [**Computer-communication Networks**]: Network Operations—*Network management, Network monitoring*; G.3 [**Probability and Statistics**]: Distribution functions, Experimental design

General Terms

Experimentation, Measurement, Performance, Security

Keywords

RF-based location estimation, wireless LAN, propagation model, wireless sniffers, calibration-free.

*The authors are with the Laboratory of High Speed Networks (RAVEL) at Department of Systems Engineering and Computer Science, Federal University of Rio de Janeiro UFRJ, Rio de Janeiro-RJ, Brazil.

MobiWac'06, October 2, 2006, Torremolinos, Malaga, Spain.

1. INTRODUCTION

In recent years Wireless Local Area Networks (WLANs) based on IEEE 802.11 standard [6] (also known as Wi-Fi) has become a very popular alternative for local area networking. It also made user mobility possible, and the proliferation of portable devices, and increasing coverage area and communication speeds introduced a new application domain. The main challenge now, in wireless networks, has shifted from speed and capacity to services, where context-aware computing became an emerging paradigm. Context, as defined in [9], is the knowledge of a user's location, activity, or goals that can be used to filter and modify the way information is presented, its content or even trigger automatic behaviors that benefit the user. The growing interest in pervasive computing and location-aware systems and services provides a strong motivation to develop techniques for estimating wireless devices positions in both indoor and outdoor environments.

Location by it self is useful, because it gives meaning to what the users are doing and what their interests are. Location Based Services (LBS), applied in the WLAN context, have been target of recent researches, once it opens perspectives for new applications, adding value to such networks. On this sort of applications, some service is offered to the user in a way that an application's input and output parameters are directly influenced by the user's physical position. For instance, automatic telephone call forwarding inside buildings, non-human tourist guides in museums, guiding costumers to a specific store in a shopping mall or permitting mobile users to print to the nearest available printer.

It is possible to deal with problems related to location and positioning in many different ways, depending on the application's desired accuracy, response time and the surrounding environment. In the present work, we propose a new system capable of detecting and locating wireless devices, with no human intervention or client side effort of any kind. We approach the problem of location estimation by using an architecture based on wireless sniffers in order to measure some radio-frequency (RF) characteristics of the WLAN signal, such as the Received Signal Strength Indicator (RSSI). Through this, our goal is to active positioning accuracy similar to previously proposed RF-based techniques, using only the existing WLAN infrastructure in the site where the location system is to be deployed, without needing any specialized or dedicated hardware.

The main focus of our work was not to bring new radio signal propagation models for location estimation or new algorithms such as triangulation. It is rather to propose a new method to build location models where classical location estimate algorithms could be intelligently aggregated and modified to improve accuracy and diminish cost.

The remainder of this paper is organized as follows. In Section 2 we survey related work in location estimation techniques. In Section 3 we discuss the variations in RSSI and the characteristics of the wireless channel. Section 4 defines two basic architectures for locations systems and describes the one, applied in our research. Our research methodology and proposed location estimator is presented in Section 5. Section 6 brings an experimental setup where our proposed methodology for location estimation was applied in a real world WLAN environment. Finally, Section 7 concludes the paper, highlighting the main contributions of our work and giving directions for future studies.

2. RELATED WORK

Several location systems based on various technologies such as infrared (IR) [1, 17], ultrasound [11] and RF signal [9, 3, 18, 13, 7, 20, 16, 14] have been proposed. Prior work in the area of location estimation could be classified into the following categories: (i) the ones that use a special dedicated infrastructure (specialized hardware) for location proposes, like the Global Positioning System (GPS) [4] and (ii) the ones that use properties of an existing communication network.

GPS is the most widely used location system for outdoor environments, but there are some drawbacks in location mechanisms of that nature. First, it does not provide good accuracy inside buildings or in the absence of Line-Of-Site (LOS) between transmitter (t_x) and receiver (r_x). Second, it requires that every WLAN device must be equipped with dedicated hardware, which increases its cost, weight and power consumption.

Traditional geometric methods for locating wireless devices are based on angle-of-arrival (AOA), time-of-arrival (TOA) or time-difference-of-arrival (TDOA). In geometric approaches the RF signal measurements are transformed into angle and distance estimates, from which the signal source location is deduced applying basic geometry and triangulation. While this techniques have been found to give good results outdoor, they are not so effective when deployed indoors, because of multipath interference. The need of specialized hardware and fine-grain time synchronization also contributes to increase the cost of this solutions.

Another approach that needs special hardware is the IR-based one. It provides accurate location information but, suffers from poor scalability due to limited IR range and high implementation and maintenance cost.

Under the class of location systems that take advantage of properties of an existing communication infrastructure, reference [3] introduced an alternative for estimating wireless device's position in a WLAN. In this method, a client device measures the amount of power it receives from an Access Point (AP) and uses this information to discover its own (x, y) location. The estimation process described in [3] is divided in *off-line* phase (also called *calibration phase* or *location fingerprinting*) and *real-time* phase. During the off-line phase, the signal strength received from several APs are measured at fixed selected locations forming a grid over the monitored area. This grid positions and its respective RSSI values are recorded and stored in a database, resulting in a *radio propagation map*. During the real-time phase, the wireless client measures the RSSI values from all the APs in range and tries to "match" it with some of the RSSI values in the propagation map in order to estimate its location. In [9, 18, 13, 7, 20, 16, 14, 2], the same propagation-map-based technique was used with different methods for matching the real-time RSSI measurements with the off-line phase ones.

However, the main problem with map-based techniques is the calibration effort in the off-line phase, addressed in [2, 15, 5, 8, 10]. The accuracy of such systems depend on this procedure, that consists of physically move a wireless device over each radio map grid point and capture RSSI values from APs. One can consider this kind of procedure to be not practical or a considerable barrier to wider adoption of such location methods. In [13] the authors reported a 4 hours calibration phase, against approximately 9 hours reported in [18].

While calibration-based efforts present good accuracy results, there is still room for performance enhancements. Due to the very dynamic nature of the RF signal, the assumption that the radio map built in the calibration phase remains consistent to the measurements performed in the real-time phase does not hold in practice. This brings the necessity of rebuilding the radio map from time-to-time. Thus, it seems more reasonable to really on a fully-automated system, capable of acknowledging RSSI characteristics and variations in both spatial and time domains, accounting it in order to build and rebuild the location model.

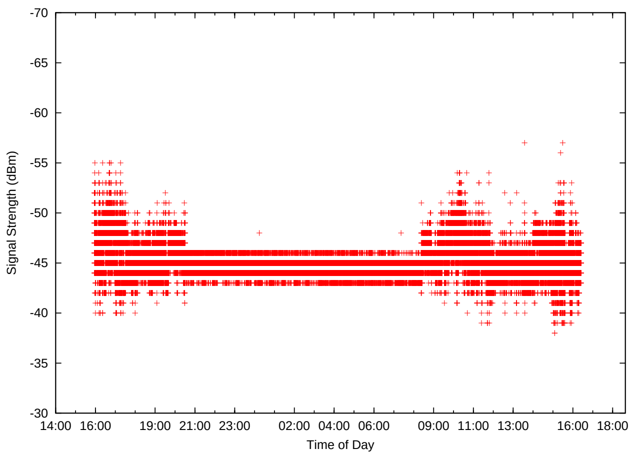

Figure 1: RSSI values measured by a wireless sniffer from an AP 10.7 meters away (with LOS), for a period of 24 hours.

3. WLAN CHANNEL CHARACTERISTICS

The focus of our work is in wireless networks based on the 802.11 standard, operating over radio frequencies in the 2.4 GHz band. In such wireless communication systems the radio channel characteristics places fundamental limitations on its performance. The mechanisms involved in signal propagation can be generally attributed to reflection, diffraction, and scattering [12]. It is pointed in [2] that in any normal WLAN environment, changing in furniture placement, surrounding structures and occupancy conditions may seriously affect signal propagation conditions, as seen in Figure 1. In this figure, we plot the RSSI measured by a r_x (wireless sniffer, defined in Section 4.2) 10.7 meters away from the t_x (a standard AP), for a period of 24 hours. In this test, the sniffer extracted RSSI information from beacon packets transmitted by an AP at a rate of 1 beacon per second. We observe that from 21:00h to 8:30h, when there was no movement of people in the office where the experiment took place, the RSSI varies only 4 dBm, against 16 dBm in busy hours.

RSSI fluctuations can be explained by two phenomenons known as large-scale and small-scale fading. The first one is caused by the separation distance between t_x and r_x, where the RSSI decreases as the distance grows. The former is given by the rapid fluctuations of the RSSI over very short travel distances or short time durations caused by multipath in the radio environment. Figure 1 illustrates small-scale fading where RSSI varies up to 16 dBm over an 1 second interval. In Section 5 we show how to minimize the effects of small-scale fading, averaging RSSI over time and using a large-scale fading propagation model with parameters determined dynamically.

4. WLAN LOCATION ARCHITECTURE

Our interest is in studying architectures for location mechanisms that uses an existing WLAN infrastructure. This is interesting in terms of cost efficiency, once no additional hardware is needed. With that in mind, we now present the components involved in the location estimation process. The nature of this components, their characteristics and the way they interact with each other is defined here as the architecture of the location system. There can be two architecture categories for location engines. The first is based on the client-server paradigm and is the most used in previous work. The second one, used only in [8] and in our work, is based on wireless *sniffers*.

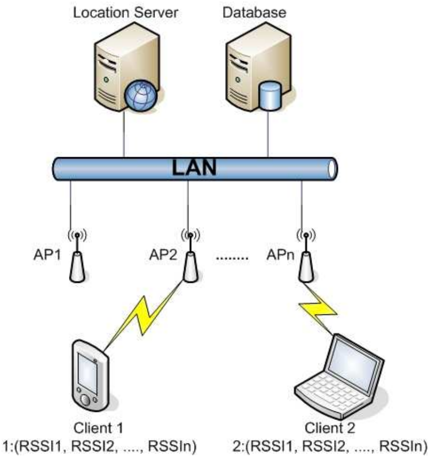

Figure 2: Client-Server-based architecture.

4.1 Client-Server Architecture

In this architecture, the position estimation process occurs in two steps, off-line and real-time phases, as described previously in Section 2. Figure 2 shows an example of a WLAN where a client-server-based location mechanism was implemented. Wireless clients are associated with one of the APs but receive signal from other APs in range. A software installed in the client is responsible for extracting from the wireless network interface, the RSSI values. Each wireless client measures the RSSIs from n APs in range and send the vector $(RSSI_1, RSSI_2, ..., RSSI_n)$ to the location server. This last one is a software that receives the RSSI vector from the client and matches it with the radio propagation map stored in the database at the off-line phase.

With the client-server approach, it is possible to enumerate some problems besides the already mentioned calibration effort. What happens to a LBS that relies upon a location mechanism based on this architecture, if the user does not start the application software for any reason? The need for the wireless client to download, install and run extra software can also be a concern in a power constrained environment [15]. Moreover, this architecture can be applied only to LBSs in which the wireless user is interested in being located. For security and management applications, a sniffer-based approach is more suitable, as it will be seen in the following.

4.2 Sniffer-Based Architecture

Here we define *sniffers* as softwares that monitor a network interface, capturing all traffic flowing through it. In our work, sniffers are entities formed by a personal computer (PC), a wireless network interface controller (WNIC) installed and configured on this PC, and the sniffer software running in order to capture traffic on this WNIC. An Ethernet NIC is also installed on the sniffer in order to setup a LAN communication between the different components of this architecture. Our sniffer software was implemented in C programming language under Linux, and in order to read RSSI information from 802.11 frames we had to put the WNIC in RF *monitor mode*, where the WNIC works only as a passive entity, unable to send packets in the wireless medium and capable of capturing every frame transmitted in the same channel (and/or adjacent channels) it is operating in.

Figure 3 shows an example of a WLAN where a sniffer-based location mechanism was implemented. In this scenario, 3 sniffers monitor the wireless medium and they are responsible for two main tasks: (1) detecting wireless client devices and recording RSSI values from it, (2) and measure RSSI from one or more reference devices in order to construct the location model. This tasks are executed simultaneously and uninterruptedly.

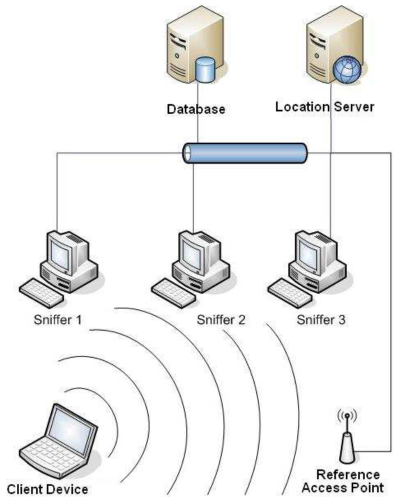

Figure 3: Sniffer-based architecture.

The first one consists in capturing $RSSI_d$ and MAC_d address from the wireless client device with indice d. In our implementation, the sniffers capture frames for each detected transmitting device during 1 second (*Capturing Interval - CI*), taking advantage of high auto-correlation between consecutive RSSI values in this interval [19]. The average value $\overline{RSSI}_d$ of all RSSIs measured in the CI is computed. This way, the tuple ($\overline{RSSI}_{1;d}$, $\overline{RSSI}_{2;d}$, ..., $\overline{RSSI}_{i;d}$, ..., $\overline{RSSI}_{k;d}$) will be sent to the database, where $\overline{RSSI}_{i;d}$ is the average of RSSI values measured by the i-th sniffer from device d during the CI, and k is the total number of sniffers in range of d.

The second sniffer task consists in capturing M *beacons* frames [6] sent by the *reference access point(s)* (AP_{REF}). The sniffers extract from this frames the corresponding RSSI values and send the pair $(\mu_{i;n}, \sigma_{i;n})$ to the database, where $\mu_{i;n}$ and $\sigma_{i;n}$ are respectively the average and standard deviation of the M RSSIs measured by the i-th sniffer from the n-th AP_{REF}. This last one, not only plays the part of a regular 802.11 AP, providing connectivity to wireless clients and to the LAN infrastructure behind it, but it also works as a reference for the location model construction. In the proposed system, the sniffer's and AP_{REF}'s positions are fixed and known, and this two components share a $1 : n$ relationship, as each sniffer is related with one AP_{REF} and one AP_{REF} is related to many sniffers.

When implementing a WLAN, the APs are generally disposed in order to active the largest possible coverage area, minimizing superpositions. This means that the larger the area to be covered, the greater the number of APs to be deployed. In client-server-based location engines, superposition is needed because wireless clients have to measure RSSIs from three or more APs to locate itself. This will increase even more the number of used APs, also increasing the cost of such network. Moreover, including extra APs in the network also means opening potential security threats and backdoors for an attacker. Thus, it is preferable to implement passive devices like sniffers.

The system component responsible for constructing the location model and estimating devices positions is the *Location Server*. It downloads from the system database all information it needs to build the model, like the fixed (x, y) positions of the sniffers and AP_{REF}(s), dimensions of the monitored local (X_{max}, Y_{max}), the used grid resolution, MAC addresses from all the devices detected by the sniffers, its $\overline{RSSI}_{i;d}$ corresponding values, and the pair ($\mu_{i;n}$, $\sigma_{i;n}$) so the location server can build the model. The construction of the model is described in the following section.

5. LOCATION ESTIMATOR

Before describing the problem of estimating the location of a WLAN device, it is necessary make some definitions. Let L be a physical bidimensional space. From each position $l \in L$, it is possible to have RSSI measured by k sniffers, given a t_x device located at l. In this work we assume L discrete. We also define a signal space S with k dimensions where each element of this space is a vector of dimension k in which its positions represent RSSI readings from the k different sniffers. Samples from the signal space S are denoted s. This way, the problem of locating a wireless device given RSSI measurements can be described as a *maximum a posteriori* problem, where, given a RSSI vector $s = (s_1, s_2, ..., s_k)$, we would like to determine the position $l \in L$, that maximizes the probability $P(l|s)$. It is possible to say that $P(l|s)$ is the probability of a t_x device to be located at l, given the RSSI vector s measured by k sniffers.

5.1 Building the Model

After reading the needed information form the database, the location server uses it to build the *radio propagation map* (RPM), in which each grid position $l = (x, y)$ is associated to a probability distribution $P(s|l)$. For each sniffer i, the location server will build a RPM in a way that, for k sniffers a total of k RPMs will be built at the server. The $P(s_i|l)$ distribution denotes the probability of a sniffer i to measure a signal strength s_i from a t_x located at a given position l. To represent $P(s|l)$, we used a Gaussian distribution (Equation 1) [3, 7, 13, 14, 16, 18, 20]:

$$P(s|l) = \frac{1}{\sigma_{(l)}\sqrt{2\pi}} \; \exp\left(-\frac{(s-\mu_{(l)})^2}{2\sigma_{(l)}^2}\right), \qquad (1)$$

where $\mu_{(l)}$ is the expected RSSI value measured at the sniffer, given a wireless t_x located at l and $\sigma_{(l)}$ is the distribution's standard deviation. To estimate $\mu_{(l)}$ we used a large-scale propagation model presented in [12] and also used in [3], defined here by Equation 2.

$$\mu_{(l)}(d) = \mu_0(d_0) - 10n_0 log\left(\frac{d}{d_0}\right) - \alpha, \qquad (2)$$

But in our work we used this model in a slightly different manner. Here, α represents the total attenuation caused by obstacles between t_x located at $l = (x, y)$ and the sniffer, $d = \sqrt{(|x_{sniffer} - x|)^2 + (|y_{sniffer} - y|)^2}$ is the distance between t_x and sniffer, the distance between AP_{REF} located at $l_0 = (x_0, y_0)$ and the sniffer is given by $d_0 = \sqrt{(|x_{sniffer} - x_0|)^2 + (|y_{sniffer} - y_0|)^2}$, n_0 is the path loss component that indicates the rate at which the RSSI decreases with the distance and $\mu_0(d_0)$ is given by $\mu_0(d_0) = \mu_{i;n}$ and represents the average RSSI value measured by the i-th sniffer from its pair the n-th AP_{REF}. The value $\sigma_{(l)}$ also changes with the position l, but here we used $\sigma_{(l)} = \sigma_{i;n}$. The value of $\mu_{(l)}$ can be estimated by the propagation model but, as the value of $\sigma_{(l)}$ can only be determined by measurements at each location l, we used $\sigma_{(l)} = \sigma_{i;n}$ as an approximation.

5.2 Rebuilding the Model

Using the propagation model parameters this way, it is possible to reconstruct the RPM every time that significant variations in the RSSI values occur (i.e. variations like the reported in Figure 1). We used the RSSI measurements between the pair sniffer/AP_{REF} to "sense" whether the WLAN propagation environment has changed and when the RPM should be rebuilt.

Assuming that RSSI follows a Gaussian distribution, average $\mu_{i,n}$ and standard deviation $\sigma_{i,n}$ can be estimated by real-time measures between the pair sniffer/AP_{REF}. Thus, it is possible to reconstruct the RPM for each pair sniffer AP_{REF} every time RSSI values measured by the sniffer i from the AP_{REF} n shows relevant statistical deviations. For instance, if m in M consecutive RSSI measurements fall outside the interval $(\mu_{i,n} + \sigma_{i,n}, \mu_{i,n} - \sigma_{i,n})$, a new RPM should be built. In that case, M more RSSI samples should be captured in order to recalculate new values for $\mu_{i,n}$ and $\sigma_{i,n}$.

An alternative scheme to reconstruct the RPM is to simply do it every T seconds. Rebuilding periodically the RPM despite of alterations in $\mu_{i,n}$ and $\sigma_{i,n}$ would keep the RPM constantly up to date so that, in the best case scenario, the map would be rebuilt even if there is no need to do so, and

$\mu_{i,n,t}$ and $\sigma_{i,n,t}$ new values would be very close to $\mu_{i,n,t-T}$ and $\sigma_{i,n,t-T}$. Great changes in $\mu_{i,n,t}$ and $\sigma_{i,n,t}$ would indicate the necessity to rebuild the RPM, and this necessity would last, in the worst case scenario, T seconds.

5.3 Estimating Device's Positions

In our proposal, we used some modifications in the methods presented in [18, 13, 20], in conjunction with our technique of sampling RSSIs from reference points over time.

As mentioned before, given a vector $s = (s_1, s_2, ..., s_k)$, we want to find a position $l \in L$ that maximizes the probability $P(l|s)$. Applying Bayes rule, we have $P(l|s)$ given by Equation 3.

$$P(l|s) = \frac{P(s|l) \cdot P(l)}{P(s)} = \frac{P(s|l) \cdot P(l)}{\sum_{l' \in L} P(s|l') \cdot P(l')}; \qquad (3)$$

where the sum goes through all the possible positions $l' \in L$ in the RPM grid. $P(s|l)$ is the probability of the sniffers to receive s from a t_x located at l. $P(l)$ is the probability *a priori* of finding a t_x in the position l. It provides an easy manner to incorporate to the location system, background information about mobility pattern and tracking. In other words, it is possible to suppose that a wireless device is more likely to be found next to tables and inside offices than inside bathrooms. $P(l)$ could be used to determine this likelihood. It could also be determined by some mobility profile and due to the fact that if a device is in a given location, it is more likely to find it in adjacent locations before a small period of time. If the user profile is unknown, a uniform probability distribution could be used to model $P(l)$.

We estimate $P(s|l)$ for each position $l \in L$, building a RPM for each sniffer, as described above. Assuming a wireless t_x located at l, k sniffers will measure a vector $s = (s_1, s_2, ..., s_k)$, and for each sniffer the probability $P(l|s_i)$ will be calculated. If we want to find $P(l|s)$, assuming that RSSI measurements s_i are independent, it is possible to write:

$$\begin{aligned} P(l|s) &= P(l|s_1, s_2, ..., s_k) = \\ &= P(l|s_1) \cdot ... \cdot P(l|s_k) = \prod_{i=1}^{k} P(l|s_i); \qquad (4) \end{aligned}$$

where $P(l|s_i)$ is given by Equation 3.

Equation 4 denotes the probability of a t_x device to be located at l, given that k sniffers observed RSSI values from this t_x, originating the vector $s = (s_1, s_2, ..., s_k)$. Figure 4 shows an example of a proability map where for each grid point is given a probability $P(l|s)$ calculated by Equation 4. This figure was generated during a real position estimation performed by our calibration-free method. The estimator's output could be the position l that maximizes the value of Equation 4. Position $l = (0, 9)$, according to Figure 4, is the position associated with the highest $P(l|s)$. This output gave us an error of 0.4 meters for this one estimation example. However, this output can be treated in order to improve accuracy. We used two techniques for outputting the estimated position of the detected wireless device: *Estimation Window* and *Center of Mass* [20].

5.3.1 Estimation Window (W)

While the sniffers measure new RSSI values from the frames transmitted by wireless devices, the probabilities in each grid point are recalculated and modified in real-time. Sometimes several different grid points could present similar values of $P(l|s)$, what could introduce error in the estimation, as the highest value of $P(l|s)$ could be in a location l far away from the device's real position. An example of this phenomenon can be seen in Figure 5. During our experiments a transmitter (notebook equipped with a WNIC) was placed at 10 different positions, chosen randomly over the monitored local. For each of this 10 positions 500 position estimations were taken by our system. In this experiments we recalculate the RPM each T seconds. We could see that cases like the one in Figure 5 (a figure like this was generated for each estimation) are common but much less frequent than cases like the one in Figure 4. This two figures were generated between an interval of only 1 second, which demonstrate the great volatility of the wireless channel and the capacity of adaptation of our system.

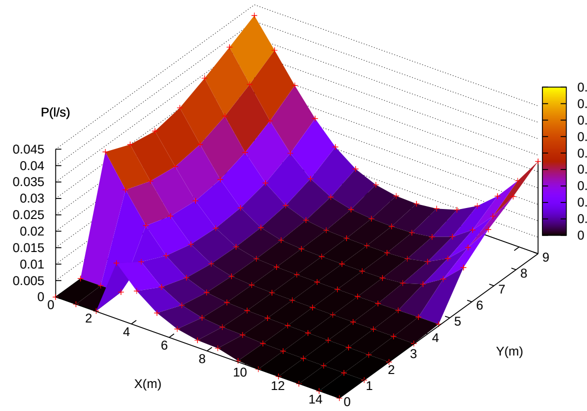

Figure 4: Probability distribution of finding a t_x device in a grid position l, calculated by the propose location engine. There is a clear distinction in $P(l|s)$ between the grid points, which allows more accurate results.

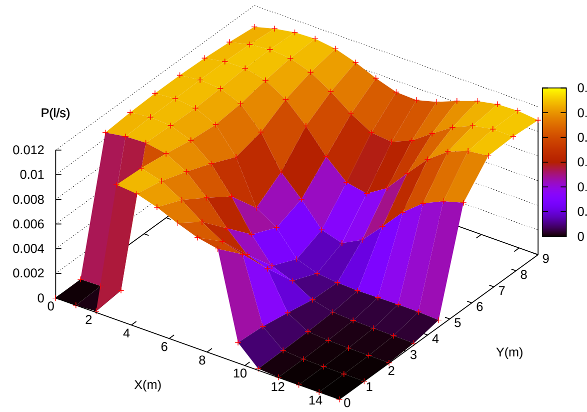

Figure 5: Probability distribution of finding a t_x device in a grid position l, where there is no clear distinction in $P(l|s)$ between the grid points. This is caused by interference in the wireless channel and allows less accurate results.

We notice that in situations where the error is big, the probability $P(l|s)$ is small, and when this probability is big, the error is small, as we can see in Figure 6. We can't say anything about the error in the case where the probability is small, but this results motivated the Estimation Window technique.

The system estimates the position l where $P(l|s)$ has the greatest value between all other positions in the RPM. This denotes one estimation. We then compute one system output[1] for each W estimations, where W is defined as the estimation window size. For instance, for $W = 10$, 10 estimations are generated. After that, the system checks the value of $P(l|s)$ for each estimation and outputs the one with the greater value of $P(l|s)$ between the $W = 10$. This method works as a filter and causes impact on the results in two aspects: increasing response time and accuracy. The first one is due to the fact that one output would take the time to generate W estimations, and the second because greater errors are normally associated with small values of $P(l|s)$ and consequently filtered, as the system chooses to output the estimation with the greatest $P(l|s)$ value within the W consecutive estimations.

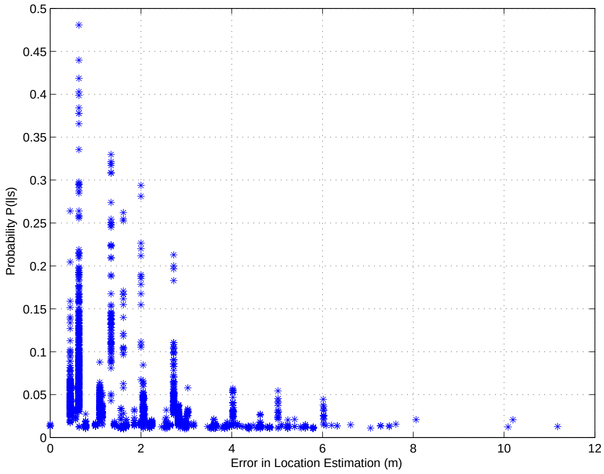

Figure 6: Probability $P(l|s)$ for the 5000 location estimations. Each estimation is associated to a probability $P(l|s)$ and a location error.

This technique can be described as a discrete estimator, as it outputs only locations over the grid points in the RPM. The following technique can be considered a continuous estimator, once it can output any coordinate (x, y), for continuous axis x and y.

5.3.2 *Center of Mass (CoM)*

The main idea behind this technique is to think of each location l as an object in physical space who's mass is equal to the normalized probability $P(l|s)$ calculated for each $l \in L$. For convenience we will make $P(l|s) = m$. Thus, calling m_j the mass of the location l_j, we can define the system output as a location Z (the center of mass) given by Equation 5:

$$Z = \frac{\sum_{j=1}^{N} m_j \cdot \overline{L}(j)}{\sum_{j=1}^{N} m_j}; \qquad (5)$$

where $\overline{L}$ is a list of all grid points in the RPM ordered in a descending order according to the normalized probability m, $\overline{L}(j)$ is the j-th element of $\overline{L}$ and N is a system parameter that indicates the number of locations taken into account in the center of mass calculations, where $1 \leq N \leq ||\overline{L}||$. Notice that for the particular case where $N = 1$, the CoM technique is equivalent to estimation window technique for $W = 1$.

[1]A system output is defined here as uploading the detected device's (x, y) coordinates to the database.

6. EXPERIMENTAL EVALUATION

We implemented the proposed location engine in a real-world indoor environment in order to evaluate and validate our calibration-free method. In the following we present the experimental testbed and the overall system performance results.

6.1 Experimental Testbed

We performed all of our experiments inside our laboratory on busy hours, during normal working days. The laboratory has a dimension of 16 meters by 10 meters, allowing a RPM grid of 160 points, with a grid resolution of 1 meter. The results presented in this paper were generated using 3 sniffers and 1 AP_{REF}. The sniffers were implemented in Linux workstations (i686 kernel 2.6.13.4), where a PCI WNIC was installed and configured to work in monitor mode along with our sniffer software. Although code was written to make sniffers to jump through 802.11 channels, for the sake of simplicity, for generating our results the sniffers kept monitoring only one channel (the same channel used by the AP_{REF} and the clients used in the experiment). We used a Cisco access point as the AP_{REF}, broadcasting beacon frames at a rate of one each 100 milliseconds. The wireless client devices used in the experiments were two Dell Latitude notebooks equipped with PCMCIA Orinoco and Enterasys WNICs. In order to constantly generate traffic, so that the sniffers could extract RSSI information from, we used the comand *ping* on the client devices.

In our experiments, we performed a total of 5000 estimations for each output technique variation (W=1, W=10 e CoM), while the client devices were positioned at 10 different positions (500 estimations each) randomly chosen. Table 1 brings a summary of the system parameter values used during the experiments.

Table 1: Parameters used in the experiments to generate the presented results.

Parameter	Value
α	0
n_0	2
T	10 seconds
Grid Resolution	1 meter
(X_{max}, Y_{max})	$(16m, 10m)$
M	60 beacons
N	5
CI	1 second
$P(l)$	Uniform

6.2 Obtained Results

In order to evaluate the proposed system's accuracy, we define the metric *location estimation error*, which is the one normally used in previous work. This error is the distance (in meters) from the point (x,y) indicated by the system output, to the real position of the wireless client device.

Figure 7 shows the ECDF (Empirical Cumulative Distribution Function) of the error in location estimation for each output technique discussed in the previous sections. The stair behavior observed in the curves for W=1 and W=10, differs from the smoother behavior presented by CoM. This happens due to the fact that the first two ones are variations of a discrete estimator, and our error metric can assume 160 different values, consequence of 160 different (x,y) pairs that the estimator can output. CoM is a continuous estimator and the error can assume an infinity of values.

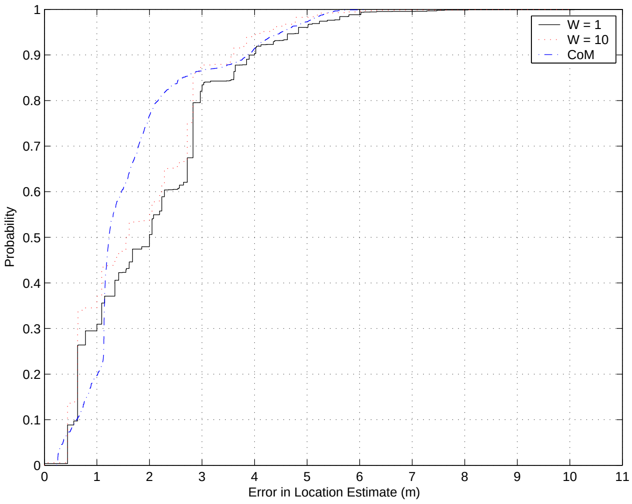

Figure 7: Empirical cumulative distribution of the error in location estimation, for each technique used in the estimator's output.

We can notice an improvement in accuracy of approximately 11% on the average, when using W=10 instead of W=1, and an even better performance on the average when using CoM instead of W=1, with a gain of approximately 18%. Estimation erro results are summarized in Table 2.

Table 2: The 50^{th}, 75^{th} and 90^{th} percentile values for the error distance (reported in meters), for each technique used in the estimator's output.

Technique	50^{th}	75^{th}	90^{th}	Mean Error
W = 1	2.00	2.83	4.00	2.07
W = 10	1.61	2.72	3.61	1.84
CoM	1.23	1.95	3.82	1.70

Figure 8, gives the location error occurrence for the three used output techniques. Notice that for errors less than or equal to 1 meter, W=10 has some advantage with 37% of the estimation erros under this threshold, against 20% when using CoM. However, 77% of the estimations presented an error inferior or equal to 2 meters when using CoM, against 54% when W=10 is used. For 77% of the estimations when W=10 is used, the presented error is less than or equal to 2.83 meters.

7. CONCLUSIONS AND FUTURE WORK

Our main objective was to propose a calibration-free, automated, RF-based method for estimating the location of WLAN devices with no need of specialized extra hardware, that does not demand software to be installed in wireless clients and that is equivalent, in terms of accuracy, to other published work in WLAN location estimation. We managed to accomplish this by using a large-scale propagation model, in which its parameters are dynamically determined over time, through RSSI measurements between the pairs sniffer/AP_{REF}.

Before setting up our experimental testbed, we did not perform any study to determine the best sniffers or APs placement in order to improve accuracy. The sniffers were implemented in workstations previously deployed in the laboratory, without moving any equipment around. The idea behind it was to show that we could accomplish good location accuracy results using an already deployed network infrastructure (wired and/or wireless), without changing any aspect of the network users routine (new software to install, new work to be done like in calibration phase or new workstations placement), and without increasing implementation costs. We searched for a complete transparent, inexpensive, off-the-shelf solution for the network user/administrator. It is part of our ongoing work to determine the impact of sniffers and/or AP_{REF} positions in the accuracy of our method. The impact of the number of devices (sniffers and AP_{REF}) used to assist location determination is also under investigation.

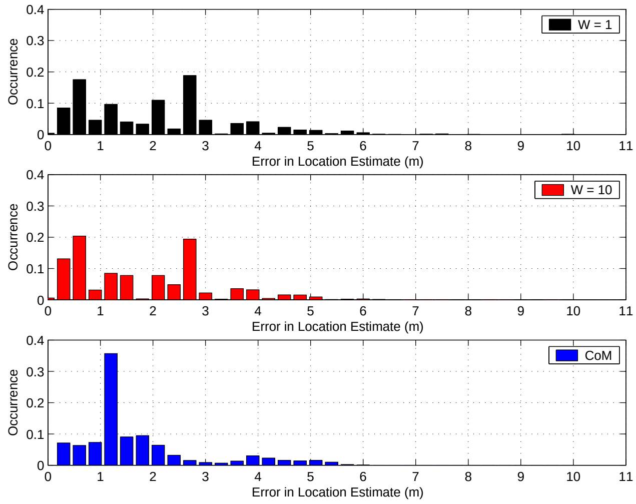

Figure 8: Normalized histogram of the error in location estimation, for each technique used in the estimator's output.

In our experiments, we used Equation 2 to estimate the average of $P(s|l)$ distribution, but we did not took into consideration the parameter α. Here we consider $\alpha = 0$. We could use different values of this parameter to tune the propagation model and approximate it even more to the reality of WLAN channels and improve our method accuracy. This is part of an ongoing research. A study of the impact of using different propagation models in location accuracy, like Ray-Trace [12] for instance, is also in progress. Moreover, we now want to determinate the optimum number M of beacon packets to be captured by the sniffers and the rate at which they are generated by the AP_{REF}, in order to provide more accurate location estimation. The same is to be done for the time T between RPM rebuilding. An investigation on how this and other system parameters would influence accuracy and determining adequate values for it, would be an interesting problem.

8. ACKNOWLEDGMENTS

This work was sponsored by FINEP, through project grant number 1956/04.

9. REFERENCES

[1] R. Azuma. Tracking requirements for augmented reality. *j-CACM*, 36(7):50–51, Jul 1993.

[2] P. Bahl, A. Balachandran, and V. N. Padmanabhan. Enhancements to the RADAR user location and tracking system. *Microsoft Technical Report*, 2000.

[3] P. Bahl and V. N. Padmanabhan. RADAR: An In-building RF-based User Location and Tracking System. *IEEE INFOCOM*, 2:775–784, Mar 2000.

[4] P. Enge and P. Misra. Special Issue on GPS: The Global Positioning System. In *Proceedings of the IEEE*, pages 3–172, Jan 1999.

[5] Y. Gwon and R. Jain. Error characteristics and calibration-free techniques for wireless lan-based location estimation. In *MobiWac '04: Proceedings of the second international workshop on Mobility management & wireless access protocols*, pages 2–9, Oct 2004.

[6] IEEE 802.11. IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, 1999.

[7] C. Komar and C. Ersoy. Location tracking and location based service using ieee 802.11 WLAN infrastructure. *European Wireless*, Feb 2004.

[8] P. Krishnan, A. Krishnakumar, W.-H. Ju, C. Mallows, and S. Ganu. A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks. *IEEE Infocom*, Mar 2004.

[9] J. Krumm and E. Horvitz. Locadio: Inferring motion and location from wi-fi signal strengths. *Proceedings 1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous'04)*, pages 4–13, Aug 2004.

[10] J. Krumm and J. Platt. Minimizing calibration effort for an indoor 802.11 device location measurement system. In *Technical Report MSR-TR-2003-82*. Microsoft Research, Nov 2003.

[11] N. B. Priyantha, A. K. L. Miu, H. Balakrishnan, and S. J. Teller. The cricket compass for context-aware mobile applications. In *Mobile Computing and Networking*, pages 1–14, Aug 2001.

[12] T. S. Rappaport. *Wireless Communications – Principles and Practice*. Prentice Hall, 2002.

[13] T. Roos, P. Myllymki, H. Tirri, P. Misikangas, and J. Sievnen. A probabilistic approach to wlan user location estimation. *International Journal of Wireless Information Networks*, 9(3):155–164, July 2002.

[14] S. Saha, K. Chaudhuri, D. Sanghi, and P. Bhagwat. Location determination of a mobile device using IEEE 802.11b access point signals. *IEEE Wireless Communications and Networking Conference (WCNC)*, Mar 2003.

[15] A. Smailagic, D. P. Siewiorek, J. Anhalt, D. Kogan, and Y. Wang. Location sensing and privacy in a context aware computing environment, 2001.

[16] A. Taheri, A. Singh, and E. Agu. Location fingerprinting on infrastructure 802.11 wireless local area networks (WLANs) using locus. *Fourth International IEEE Workshop on Wireless Local Networks*, Nov 2004.

[17] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 40(1):91–102, Jan 1992.

[18] M. Youssef and A. Agrawala. WLAN location determination via clustering and probability distributions. *IEEE PerCom*, February 2003.

[19] M. Youssef and A. K. Agrawala. Handling samples correlation in the horus system. *IEEE Infocom*, Mar 2004.

[20] M. Youssef and A. K. Agrawala. The horus wlan location determination system. *Proceedings of the Third International Conference on Mobile Systems, Applications, and Services - MobiSys*, June 2005.