A System for LEASE:

Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks

P. Krishnan[*], *A. S. Krishnakumar*[*], *Wen-Hua Ju*[*], *Colin Mallows*[*], *Sachin Ganu*[#,1]

[*]Avaya Labs Research
233 Mt. Airy Road, Basking Ridge, NJ 07920, USA

[#]WINLAB, Rutgers University
73 Brett Rd, Piscataway, NJ 08854, USA

Email: {pk, ask, whju, colinm}@avaya.com, sachin@winlab.rutgers.edu

Abstract—**In this paper, we present LEASE, a new system and framework for Location Estimation Assisted by Stationary Emitters for indoor RF wireless networks. Unlike previous studies, we emphasize the deployment aspect of location estimation engines. Motivated thus, we present an adaptable infrastructure-based system that uses a small number of stationary emitters (SEs) and sniffers employed in a novel way to locate standard wireless clients in an enterprise. We present the components of the system and its architecture, and new non-parametric techniques for location estimation that work with a small number of SEs. Our techniques for location estimation can also be used in a client-based deployment. We present experimental results of using our techniques at two sites demonstrating the ability to perform location estimation with good accuracy in our new adaptable framework.**

Keywords- Location estimation, enterprise wireless, 802.11, deployment, adaptation, sensor networks, experimentation with real networks/testbeds, system design.

I. INTRODUCTION

With the increasing use of wireless networking, especially 802.11-based wireless systems in enterprise networks, the thrust now is to develop services that provide more than untethered network access. An important class comprises those services that use end-user location information. Such services include location-aware content delivery, emergency location, services based on the notion of *closest resource*, and location-based access control. Techniques that can estimate location in indoor environments, preferably without client changes, are important to enable such services in enterprises. Traditional GPS methods cannot be used for this location estimation since they have problems working indoors.

In typical wireless deployments in an enterprise, a site is served by several access points (APs). Client devices associate with an access point to obtain connectivity. Some services (e.g., services based on the notion of closest resource) might work with a gross estimate of user location, such as to which AP the user is connected. However, other services require more fine-grained estimates of location.

Indoor wireless LAN (WLAN) location estimation can employ one of several physical attributes of the medium for estimation. The typical features that might be used are: the received signal strength (RSS) of communication, the angle of arrival of the signal, and the time difference of arrival. Among these, RSS is the only feature that is measurable with reasonably priced current commercial hardware. Related efforts [4, 5, 7, 14, 15, 19, 20, 25] have used RSS for location estimation and concentrated on locating a user in two-dimensional space, e.g., a point on one floor of a site. They have also demonstrated the viability of using RSS for the location estimation problem.

Most previously published techniques for WLAN location estimation (discussed in more detail in Section II) operate in two phases: a model building *offline phase*, and an *online phase* when estimation is performed. The model constructed in the offline phase is, essentially, a map of signal strength behavior at the site. The model is either constructed using many measurements at the site, or is parametric and depends on several variables like type and number of walls and other signal obstructors. In the online phase, a set of signal measurements is mapped to a location after "consulting" the model. The accuracy of estimation depends on the technique used to build the model, and the algorithm used to match the measured signal strengths to the model.

The deployment of location estimation systems can be done in one of two ways: in a *client-based* deployment [4, 5, 29], the client, in the online phase, measures the signal strengths as seen by it from various APs. This information is used to locate the client. The cost to an enterprise for such deployments is the cost of managing the client devices, profiling the site, building the model, and maintaining the model. In a traditional *infrastructure-based* deployment [7, 28, 30], the administrator deploys simple sniffing devices that monitor clients and the signal strength from clients. This sniffed information is used to locate the clients in the online phase. The cost to enterprises in such deployments is the cost of deploying the necessary hardware and software, the time and effort to build the model (if it is not completely automated), and the cost of maintaining the model (if it cannot be automated). For ease of management, including provisioning, security, deployment and maintenance, we believe that enterprises would prefer an infrastructure-based deployment, especially if building and maintaining the model

[1] Portions of this work were done when S. Ganu was visiting Avaya Labs Research.

can be automated. Client-based systems [16, 20] may offer more privacy than infrastructure-based systems.

Most previous work either has concentrated on pure client-based solutions that require no additional site hardware, basic infrastructure deployments that use sniffers, complex custom hardware for location estimation, or has not addressed the issue of deployment and maintenance. Another important issue that has not been studied extensively is the cost and complexity of building and maintaining the model. As pointed out in [5], even in normal office environments, changing environmental, building, and occupancy conditions could affect signal propagation models. Static models may not be suitable for other dynamic location estimation environments like warehouses and malls. Infrastructure changes, e.g., adding, removing or moving APs, may also require model rebuilding. Using purely static techniques for building models make the models difficult to maintain and update. As is also pointed out in [20], techniques that profile the site extensively involve a steep upfront cost and effort to deployment, and add significantly to the complexity of maintaining the model. In this context, simple non-parametric models are preferable that can be built with little or no profiling and achieve location estimation accuracy comparable to techniques that profile the site extensively.

Over the past few years, chipsets for several wireless (e.g., 802.11-based) devices have become very cost-effective. An adaptive infrastructure-based solution that can be built with off-the-shelf components and chipsets would be significantly beneficial. In this paper we present such a system, LEASE, that uses stationary emitters (SEs) and sniffers in an interesting way to provide an infrastructure-based location estimation solution for enterprises. The sniffers in LEASE can additionally be used for other administrative and management applications like security monitoring [28, 30], QoS measurements [9], etc., allowing their cost to be further amortized. The deployment model of LEASE, as we will see, allows for quick bootstrap and self-updating, helping to solve many of the problems discussed earlier.

Our deployment model motivates a location estimation problem with a twist: expressed in terms of previous work in this area, the problem translates to being able to do location estimation with accuracy but minimize the amount of profiling. Instead of using radio propagation models that are inherently parametric, we present in this paper new non-parametric modeling techniques for building our model used for location estimation, and simple decision methods for floor determination. We present detailed experimental results with our technique at two multi-floor sites, and show that users can be located with accuracy comparable to other published techniques that extensively profile the site. Additionally, our experiments use larger sites than reported in earlier literature. We also introduce the notion of an *effective normalized error* metric that models many aspects of the location estimation problem and evaluate the performance of different approaches to location estimation using this metric.

II. RELATED WORK

As pointed out in Section I, related work in the area of indoor location estimation has concentrated mostly on algorithms for location estimation as opposed to the deployment and maintenance criterion. The location estimation techniques in a wireless network can be broadly classified based on the methods used to build models and the methods used to search the models in the online phase. For building models, most techniques profile the entire site and collect one or more signal strength samples from all visible APs at each sample point. The collected information is the model where each point is mapped to either a signal strength vector [4, 14, 15, 19] or a signal strength probability distribution [6, 18, 22, 25]. Such profiling techniques require considerable investment in building the model. Furthermore, if the environment changes [5], the profiling will need to be repeated. Alternatively, a parametric model that uses signal propagation physics and calculates signal degradation based on a detailed map of the building, the walls, obstructions and their construction material, has been proposed [17] and used for location estimation [4]. Obtaining detailed maps of the building and its changes over time is, however, a hurdle that needs to be overcome for the use of this method.

In the online phase, nearest neighbor based methods [4, 15] or probabilistic techniques to match a presented signal strength vector to the model [22, 25] are used.

The complexity of building the model was identified in [5, 20]. It was tackled to some extent in [20] where a specific circularly symmetric functional relationship between signal strength and distance was generated empirically for their site. In practice, the measured signal strength contours are usually anisotropic. Our approach for modeling the signal map, as described in Section IV.A, is very different and automatically models artifacts of buildings like corridors and office areas. In [20], the authors emphasize a client-based location model and raise interesting privacy issues in location-based services. We expect that in enterprises, based on current privacy policies used for other electronic transmissions like email and web-access, the preference would be for an infrastructure-based solution [28, 30]. If privacy is desired, in our case, on entering a site a client device could download the model for that site and use it to determine its own location. As mentioned in [20], client-based approaches must also be concerned about the power requirements on the client devices that are inherently power constrained.

Sniffing for clients to provide an infrastructure-based system has been proposed [7, 23, 24, 28, 30]. Pure sniffing [7, 28, 30] does remove the dependence on specialized client software, but does not allow for self-adjusting models, something we achieve with the use of our SEs and sniffers and a location technique designed to operate under these constraints. Custom sensors have been used for location estimation in other interesting ways [16, 23, 24]. In [23] and similar systems, infra-red (IR) wireless technology is used; IR technology has limited range and hence has not become very popular. In [16], a decentralized (client-based) approach using time difference of arrival between ultrasound and RF signals from custom sensors is used for location estimation. The system in [24] uses expensive custom RF-based hardware for location estimation, and an approach based on time difference of signal arrival, which is inherently more expensive to measure. In contrast, our self-adjusting system, LEASE, is easier to bootstrap, is based on RSS and can be built with off-the-shelf components. Recent advances in sensor technology [11] and projected decrease in the manufacturing cost allow us to provide a cost-effective solution in the LEASE system.

We would like to acknowledge that in some scenarios it is possible that a client-based approach with full profiling is adequate (e.g., smaller sites with conditions where the model rarely changes). Our emphasis in this paper is on enterprise sites that require easier manageability, the ability to update models automatically, and to achieve location estimation from an infrastructure deployment.

III. The LEASE System

The LEASE system employs three main components: the stationary emitters (SEs), the sniffers, and the location estimation engine (LEE). In Figure 1 we show a possible office site floor with some access points (APs), SEs and sniffers. The location estimation engine can be anywhere in the network.

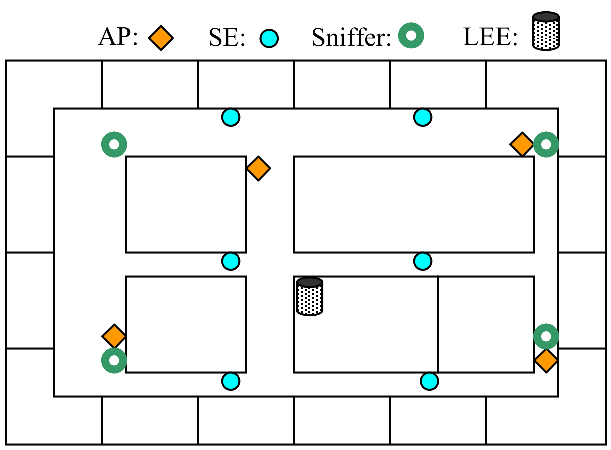


Figure 1. A possible office site with components of the LEASE system.

The SEs in the LEASE system architecture are standard, inexpensive wireless transmitters that send a few packets occasionally. The packets are meant for the sniffers, as described below in this section. These SEs have very small form factors and could be battery operated. These SEs do not need network connectivity either. At a few packets every hour, for example, the duty cycle for such SEs is extremely low. The SEs must, clearly, use the same RF technology as the clients being located.

The sniffers sniff on the wireless medium, cycling through the available (or, required) frequencies, and listen for all communication from wireless clients and SEs, recording the RSS from the clients and the SEs. The recorded information is sent to the LEE. The LEE, as we will see later in Section IV.A, also needs the coordinates of the SEs. The SEs can be configured with their location identity as an (x, y) coordinate, before being deployed at that location and the packets sent by the SE could include its coordinate information in the payload. Alternatively, the identity of the SE (e.g., its MAC id) can be mapped to the SE's location via a table maintained at the LEE or the sniffers. We expect the sniffers to be fewer in number, presumably on the order of the number of APs. The sniffers do not have to be co-located with the APs. We assume that the sniffers are connected to a power source and have network connectivity to the LEE. This network connectivity could be through a wired medium, or the sniffers could operate as a wireless ad-hoc network. The sniffers could also act as a transparent bridge between the AP and the rest of the network. Note that the sniffers can additionally perform the tasks of an SE in LEASE, and so the total number of SEs in the system is really the number of pure SEs plus the number of sniffers deployed.

A detailed description of the LEE and our techniques for location estimation appear in Section IV. Broadly speaking, the LEE does two activities: i) it collects from all sniffers the RSS from the SEs and the coordinate information of the SEs (or their ids and maps the ids to coordinate information). The LEE uses this information to build, and refine if needed, its signal strength model for location estimation, and ii) when a client needs to be located or tracked, the LEE uses the signal strength information from the client as recorded by all the sniffers, in conjunction with the model, to locate the client.

We note that the terms SEs and sniffers as used in the LEASE system may have some similarity to sensors [11] and clarify the possible relationship here. The SEs as they appear in LEASE are also small, lower-power devices like sensors, but do not "sense" anything specific. They could, however, record information such as the noise level at their location, and pass it along in their transmitted packet. More importantly, if sensors are already deployed in an enterprise for other purposes, these sensors can serve the purpose of SEs for LEASE. We would like to emphasize that all that the SEs are used for is to transmit a few packets every so often. The detection of the packets at the sniffers allows RSS to be measured, which is what the LEASE system needs. Additionally, the sniffers in LEASE look like "traditional" sensors; however, they are more communication-intensive, require more power and may never transmit wirelessly, depending on the deployment requirements. Notice that, given our emphasis on easier deployment, with our SEs and the LEASE system, we have deliberately not introduced the notion of antenna orientation as in [4]. However, the system model does not limit one from using mechanical techniques to continually re-orient SEs and pass the orientation information in the payload of the SE's packet, or place multiple SEs at a location with oriented antennas.

We now make some interesting observations regarding the LEASE system and previous client-based location estimation techniques.

A. Observations Regarding the LEASE System

A fundamental aspect of signal propagation taken for granted is the *reciprocity* of received signal strength [13]. Specifically, if stations A and B transmit to each other with equal power, the RSS as seen by A for transmissions from B is essentially the same as the RSS as seen by *B* for transmissions from *A*. (This is not true for other local metrics like noise.) We further assume *visibility*, i.e., when a client needs to be located, packets sent by the client are viewed at the sniffers within range. We can elicit a transmission from a client via IP-based methods (e.g., a ping), or sniff the acknowledgment frames for MAC-layer transmissions. Visibility is necessary for infrastructure-based deployments.

Let *S* be the set of SEs and let *N* be the set of sniffers in LEASE. Let *A* be the set of access points. The set *A* includes all APs used by clients in their location estimation computation. Let *L(S), L(N)*, and *L(A)* denote the set of locations of the SEs, sniffers and access points respectively.

Observation 1. Consider a client-based technique *CT* that only uses RSS for location estimation, and that uses model *M(t)* for its location estimation at time *t*. Reciprocity and visibility imply that the LEASE system with $S=\phi$, where the LEE uses model *M(t)* for its location estimation at time *t*, is equivalent to *CT* if $L(N) = L(A)$.

Observation 1, intuitively understood even before, implies that the LEASE system can provide an infrastructure-based deployment without any SEs for currently proposed client-based techniques.

We now make a stronger observation about adaptive models. Let *X* be the set of points at which a profiling-based technique collects RSS samples to build its model for location estimation, and let *L(X)* be the set of locations of these points. Let *T* be the set of times at which RSS values are computed to build and rebuild a model, e.g., as suggested in [5]. We assume, as in prior work that the profiling techniques only measure RSS from the APs in *A*. This leads to the following observation.

Observation 2. Reciprocity and visibility imply that the LEASE system is at least as good as a profiling-based system when $L(S) = L(X)$, $L(N) = L(A)$ and the SEs in *S* transmit packets at time $t \in T$.

Observation 2 implies that deploying the SEs and sniffers provides an easy way to build adaptive models. The model adaptation time interval affects the duty cycle and hence the power consumption of the SEs.

Since parametric location estimation techniques have inherent problems with maintenance and bootstrapping, we concentrate on non-parametric models. The novel deployment and adaptation model of the LEASE system presents an interesting location estimation problem: can we develop non-parametric techniques for location estimation that do not compromise on the quality of location estimation but minimize the number of SEs required. Based on Observation 2, we can also cast the problem as follows: Can we develop a profiling-based location estimation technique that reduces the amount of profiling needed? We discuss this issue and present our LEE in the next section.

IV. LOCATION ESTIMATION IN THE LEASE SYSTEM

The location estimation engine (LEE) in the LEASE system builds and refines its model based on RSS as seen at the sniffers for packets received from a few SEs that are placed at known locations. When a client needs to be located, the LEE is presented the RSS for a client as seen by the set of sniffers. The LEE estimates the client's location by "matching" the presented RSS vector with the current model. We now discuss the method used to build the model, the criterion for rebuilding the model, and the method used to match a client's RSS vector with the model. Another issue we discuss is where to place the SEs and sniffers.

A. Building the Model in LEE for a Floor at a Site

The traditional physics-based technique to model signal strength has been to understand variation of RSS with distance from the signal source. Fundamentally, this approach is complicated in indoor environments due to multi-path propagation and signal attenuation due to obstructions such as walls. We use a propagation model-agnostic approach to the modeling problem. We cast the problem of mapping the signal strength at the site as a pure data-modeling problem where the RSS from the SEs provide a sample of the data set.

We model the RSS as a function of the coordinates of the SEs at a site. This allows us to automatically handle any anisotropy that may be present in the data.

We build a model for each sniffer independently as follows. The first step is to smooth the data points, e.g., using a generalized additive model (GAM) [12]. In some cases, e.g., when we have very few SEs that are far apart, it might not be necessary to smooth the data and we could skip this step.

In the second step we generate a synthetic model. This is motivated by our interest in allowing our system to be able to use many of the intuitions from previously published literature for matching a client's RSS to the signal model in the online phase [4, 6, 15, 19, 20]. This includes both deterministic and probabilistic matching techniques. We divide the site into small grids (e.g., grids of 3ft × 3ft cells). Using Akima splines [1, 2, 3], we interpolate the smoothed values obtained from the GAM to estimate the RSS at each grid center. The Akima spline interpolation technique [1, 2] does a bivariate interpolation and is a local, triangle-based technique with many desirable properties including local containment of discontinuities. Our synthetic model for the specific sniffer is the generated RSS-grid information with an estimated RSS for each grid point. Notice that the location of the sniffer is not needed to compute our model, only the coordinates of the SEs and the RSS from the SEs.

We repeat the above technique for each of the *n* deployed sniffers. At the end of this process, we have a set of grids for the site, where each grid has an associated *n*-vector of estimated RSS. From the point of view of previous work, this *n*-vector corresponds to the profiled RSS from each AP as seen at each grid point, assuming the APs and sniffers are co-located.

When building the model, we use data from all SEs. In particular, if we do not see a signal from an SE, we peg that SE's reading at some small value (e.g., -92 dBm). Our modeling technique automatically uses and folds this pegged value into the synthetic model that is built. We believe and notice in our experiments that this *confirmed* absence of a signal is useful in location estimation since, intuitively, this absence does narrow down the search area by indicating that the point in question is far away from the sniffer.

Prior experimental work has concentrated on situations when a client sees all APs involved in the experiment. Usually, this has meant that at least three sniffers can see every point, or, alternatively, every point can see at least three APs. While such a deployment helps, most sites may not be engineered this way. We note in this context that most network managers would be more comfortable deploying additional (mostly passive) sniffers to facilitate location estimation rather than deploy temporary APs or relocate APs and deal with channel overlap issues. From the viewpoint of a network administrator, a sniffer is just like a (slightly special) client, and hence, easier to manage.

B. *Re-building the Model in LEE*

Previous work and our measurements clearly verify that RSS at a point follows a log-normal distribution with a standard deviation σ that can be estimated for the site by the sniffers. We rebuild the model when at least one sniffer observes an RSS from any SE that consistently exhibits a statistically significant deviation. The deviation is measured during every measurement cycle. Note that, with this approach, a model will be rebuilt if an SE or sniffer is moved, or the environment changes significantly.

C. *Locating the Client in the Online Phase*

In our current approach, to locate a client in the online phase, we use the RSS for the client from all sniffers first to map the client to a floor at a site. We then match the RSS as seen by the sniffers on the mapped floor to the model for that floor to locate the client.

1) *Mapping the Client to a Floor at a Site*

Since a typical floor attenuates a signal significantly, it has been traditionally assumed [5] that determining the floor at which the client is located should be straight-forward. We employ the following simple majority logic-based heuristic for locating the floor where the client is located: Sniffers are grouped by floor. On being presented a vector of the RSS from the client at all sniffers, we sort the vector and use a modified majority logic. We find the smallest m such that the following holds: i) Majority Rule: A majority of the sniffers from which we see the m strongest signals are in the same floor F, and ii) Stability Rule: Adding $f(m+1)$ dB (in our case, we use $f(.)$ = 5dB) to the $m+1$th signal does not change the decision. We declare the client to be in floor F. We refer to the quantity m as the decision depth.

2) *Matching the Client RSS Vector to the Floor's Model*

We use two variations of the nearest-neighbor algorithms for matching the received client RSS vector to the model: the *Full-NNS*, and the *Top(k)-NNS*. Nearest neighbor techniques have been shown to work well in prior work [4, 15]. Note that our synthetic model generation approach was used to generate a model for which known techniques for model matching can be used more easily.

Full-NNS. In this technique we match the entire RSS vector as seen from the client to the RSS vectors at each synthetic grid point to find the closest match.

Top(k)-NNS. In this variation, we consider only the top k RSS from the client. Let these k RSS values correspond to sniffers $n_1, n_2, \ldots, n_k$. We match the client's RSS k-subvector with the corresponding k-subvector of only those grid points where the sniffers with the top k signal strengths are $n_1, n_2, \ldots, n_k$. The quantity k is typically small, e.g., 3. Clearly, Top(|N|)-NNS = Full-NNS, where N is the set of sniffers.

Intuitively, if the absence of a signal is not very useful, or only the top 3 RSS at a location make a difference in location estimation, then Top(3)-NNS would perform comparably to (or better than) Full-NNS. We note that many other techniques proposed in prior work can be used or adapted to locate the client in the online phase in the LEASE system.

D. *Placing the SEs and Sniffers*

1) *Determining the Number and Placement of SEs*

Having at least a certain number of SEs and where we place the SEs are important issues. Recall from Section III that sniffers can also double up as SEs.

The SE placement and number of SEs needed are, intuitively, related to the technique used to build the model. Our model is built using a combination of smoothing and two-dimensional Akima splines as described in Section IV.A. For a bi-cubic spline approximation, at least seven points are needed for reasonable smoothing. This implies that a sniffer should see at least seven SEs. (More than one sniffer may see a given SE.) In general nonparametric regression estimation, the best possible convergence rate is attained when the design points are equidistributed [21]. We use an engineering solution to the problem of SE placement in this paper. In our case, we divide the floor of the site into approximately equal area sub regions such that at least seven sub regions are in the range of a sniffer. We install an SE as closely as possible to the center of the sub-region. In Section VI, we provide experimental results that shed some intuition on how the accuracy of estimation varies as a function of the number of SEs.

If smoothing is done using GAM, it helps if the SEs present a large number of distinct x and y coordinates; something that can be achieved with simple subdivision and perturbation techniques, or line sweep methods. An interesting optimization problem in this context is to optimize the number of SEs while keeping them as far apart as possible and maximizing the number of x and y coordinates presented. Further discussion of this problem is beyond the scope of this paper.

2) *Placing the Sniffers*

From an engineering and deployment point of view, it makes most sense to first deploy the sniffers at locations in the site where there are access points. The AP locations are also typically ones that have both power and network connectivity. One or two additional sniffers, or sniffers not collocated with the APs might sometimes be needed to ensure that sniffers, for example, are not collinear. Note that if the APs are later moved for coverage reasons (or additional APs are added for QoS reasons), the sniffers do not have to be changed or moved.

However, moving the sniffers or SEs will automatically create new models for the site.

V. EXPERIMENTAL METHODOLOGY

In this section, we present our initial observations in building a prototype of the components of the LEASE system, and the experimental methodology used to study the location estimation engine (LEE) in the LEASE system.

A. Prototype and Initial Observations

We built a prototype of a sniffer on an embedded platform [27] running Linux, using standard publicly available wireless extensions [31] with our own modifications. General 802.11-based sniffers [26] for Linux are quite popular. While the sniffers can also be adapted for use as SEs, we propose that simpler devices based on other sensor-related platforms [11] can be adapted to serve as SEs in the LEASE system.

We experimentally verified the reciprocity and visibility (see Section III.A for a definition of these terms) of signal strength. For visibility, we observed that the sniffers need to dwell on a channel for some amount of time (e.g., 350-500ms) to sense a reasonable number of terminals per channel. To measure the signal strength of a terminal/SE's transmission, a sniffer needs to receive one (or preferably more) packet(s) while dwelling on that channel. A reasonably chatty terminal will automatically satisfy this requirement. In particular, we noted that if the sniffer cycles through 11 channels, even four ping packets sent by the terminal spaced one second apart were enough for recording the terminal's presence and its RSS. In practice, a sniffer may cycle through only the three non-overlapping channels used in the enterprise. Our SEs would send several packets in a short space of time before sleeping until the next transmit cycle and will easily satisfy the visibility requirement. We expect that this activity would require relatively low duty cycle and hence will not be a drain on the SE's battery. More information related to the practical issues in designing and deploying the sniffers is discussed in [8].

B. Experimental Study of the LEASE System's LEE

An important aspect of the LEASE system is its location estimation engine. Our experimental studies are designed to evaluate the quality of location estimation of LEE and its dependence on the number of SEs used.

For our experimental study, we use the intuition from Observation 2 in Section III.A to perform a client-based study that lower bounds the performance of the LEASE system's LEE. We note that the adaptability, self-configuration, and bootstrapping capabilities of the LEASE system are significant contributions; however, the data presented below in our experiments captures the raw location estimation capability to better allow us to compare the tradeoffs with existing literature. We will see from our experiments that we can achieve accurate estimations with very few SEs.

1) Data Collection Methodology

We collected RSS data from two floors at two sites, referred to in this paper as BR and CA. Both the BR and CA sites have deployed an 802.11b wireless network. We extensively profiled one floor from each site: a map of this floor for each of the two sites is shown in Figure 2. We also profiled in a limited way a second floor from each site. The profiled information was used in our (off-line) experiments to validate our methods and draw inferences. The main location estimation experiments reported here use the information from the floor of each site that was profiled extensively; we present more information about these floors below. The data from the second floor was used to test floor discrimination; the floor plan and dimensions for this other floor are similar to the corresponding extensively profiled floor shown in Figure 2. We note that the profiling we did is not required in the LEASE system, which enables an automated and adaptable approach to location estimation; we did the extensive profiling to collect data to experimentally study several questions of interest (e.g., the impact of the number of SEs on location estimation accuracy) and report our results in this paper.

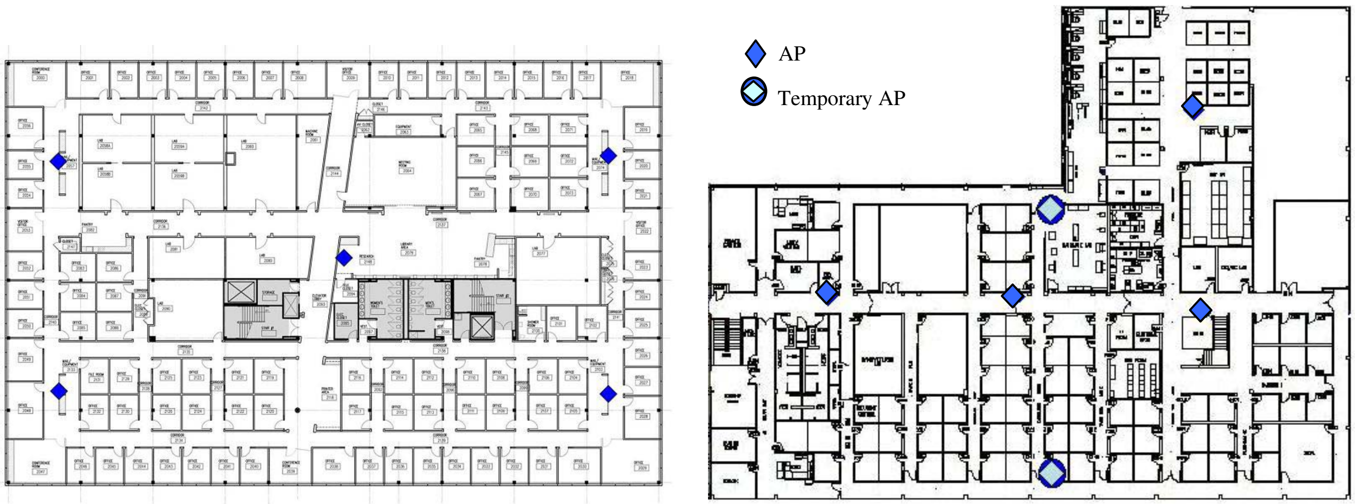
To make our RSS measurements, we used a Linux IPAQ with a modified driver updated to scan for APs. The IPAQ had a custom client and a standard Konqueror web browser. The user making RSS measurements clicked on their current location in an image of the floor as displayed on the browser. The posting of this information triggered an RSS measurement request at the client from the web server on a separate TCP channel. The web server then recorded the coordinate and RSS vector information at that location. We did not specifically orient the IPAQ in any way while taking measurements.

The BR site has 5 APs and measures 225 ft × 144 ft. We made 259 RSS measurements along the corridors of this site, and 119 measurements inside the offices and labs. The measurements were made over different sessions spanning several days.

The CA floor has 4 APs, three of which are collinear, and measures 250 ft × 175 ft, with a "slice" removed. Due to the collinearity of the three APs, we installed two temporary APs configured such that the clients could see the beacons from the temporary APs but not be able to associate with them. At CA, a colleague took 150 measurements along the corridors.

2) Experimental Methodology

We used the collected data in the following way. We classified the entire set of data points, P, into two subsets: corridor data, C, and office/lab data, O, where $P = C \cup O$. In the case of the CA data set, $O = \phi$. From the points, we chose a set $M(k)$ of k points to build the model and checked the model on the other points to compute error. Most of our reported experiments were run using what we call in this paper as the *typical case*, where the k points in $M(k)$ were chosen from C, and the model was tested against all points in P. The reason for this choice was based on where we thought the SEs are most likely to be deployed: the typical case assumes that the SEs are likely to be placed in corridors. The points in $M(k)$ were chosen by dividing the site floor uniformly into k grids and selecting the points from C that were closest to the center of the grids. Our technique for building the model is described in Section IV.A. We implemented our model building technique in Splus [32].



(a) Floor plan of site BR

(b) Floor plan of site CA

Figure 2. Floor plans for sites BR and CA showing the APs.

For comparison purposes, we also ran experiments with three other combinations we call the *average case*, the *worst case*, and the *disjoint typical case*. In the average case, the *k* points in *M(k)* to build the model were chosen from *P*, and the model was tested on all points in *P*. In the worst case, *M(k)* was chosen from *C* and the model was tested with all points in *O*. In the disjoint typical case, *M(k)* was chosen from *C*, and the model was tested on all points in *P–M(k)*. In the worst case and the disjoint typical case, the test set and the set used to build the model are disjoint.

Notice that when interpreting our experiments in the context of the LEASE system, the *k* points will correspond to the locations where SEs will be placed and the sniffers are placed at the APs. In traditional client-based estimation techniques, the *k* points will correspond to the places where RSS values are measured to build the model. Since we are mostly interested in the situation when *k* is small, we should expect that the set used to build the model, even if it is in the set used to test, should hardly affect the results.

VI. Results of the Experimental Study

We present the results from our experimental study under various categories. Recall that the number of points used to build the model is *k* and corresponds to the total number of SEs. We compute the 25-percentile, median, 75-percentile and mean error for various cases. While we are especially interested in the case when *k* is small in relation to the size of the site, for understanding the behavior with increasing *k*, we vary *k* up to ≈100 in our results presented below.

A. Metrics as a function of k for the Typical Case

In Figure 3, we plot the error metrics as a function of *k*. We make a few interesting observations from this figure. First, the error metrics (mean, median and 75-percentile) shown in the figure are all small even with small *k*. The median error goes down significantly with increasing *k*. For example, with just 12 SEs, we can achieve a median error of 15ft (4.5m) and with 104 SEs, we get a median error of just 7ft. This implies that the LEASE system can provide very good location estimation with few SEs.

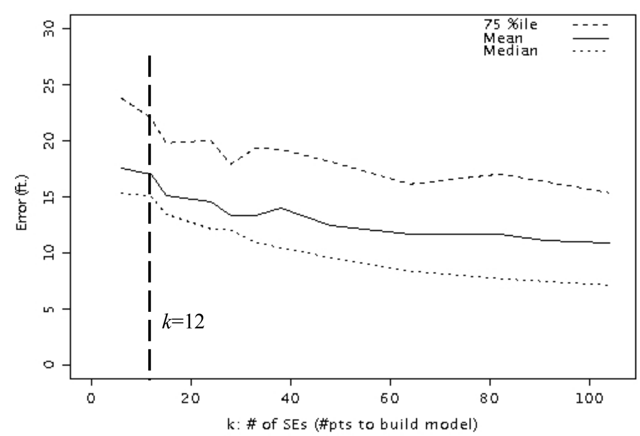


Figure 3. Metrics as a function of *k* for the typical case

B. Comparing the Various Cases: typical, average, disjoint-typical and worst-case

In Figure 4, we plot the median error for the typical, average, worst-case and the disjoint typical cases. (See Section V.B.2 for a definition of these cases.) We observe that the error variation with *k* is very similar for the typical and average cases. We expect that our method for choosing the *k* points from the data set to build the model in both cases resulted in virtually the same set of points getting chosen. We further observe, as expected, that for small k (e.g., $k \leq 50$), the typical and disjoint-typical cases are mostly identical. The worst case has a median error of a few feet more than the other cases. This is not surprising because the points used to build the model in this case are in the corridor, but all test points are inside offices (along the periphery) or in the labs. We expect that reducing the extrapolation in our technique by choosing points more along the edges to build the model might help in this case.

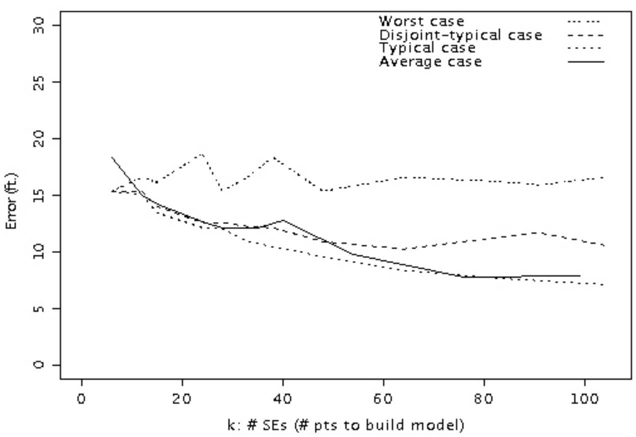


Figure 4. Comparing the various cases.

C. Comparing across sites: BR vs. CA

In Figure 5, we show how the median error for the typical case varies with k for our two sites. We notice that the general trend is similar. However, the CA site shows significantly lower median errors with increasing k. Intuitively, the model error is zero at the points used to build the model and increases as one moves farther away from them. Recall that all the data points collected in CA were from the corridors, so the model was built and tested on corridor points, which might account for the smaller errors. In the BR case, where the office points are farther away from the model points we see more error.

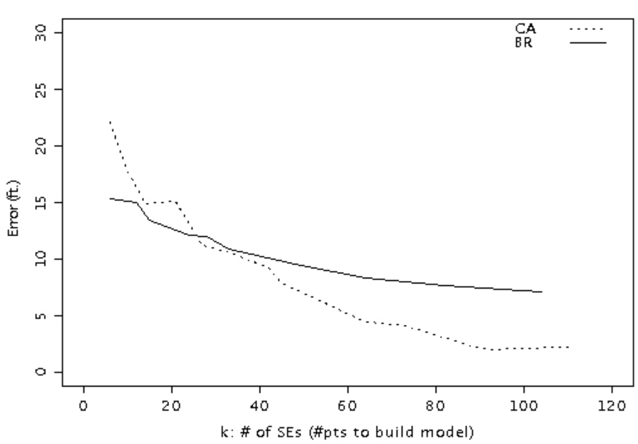


Figure 5. Comparing the two sites, BR and CA.

D. Bias in Errors along the x and y directions

We investigated if our estimates were biased. In Figure 6, we show the error distribution in the x and y directions for the case k=28. We observe no significant bias in either direction.

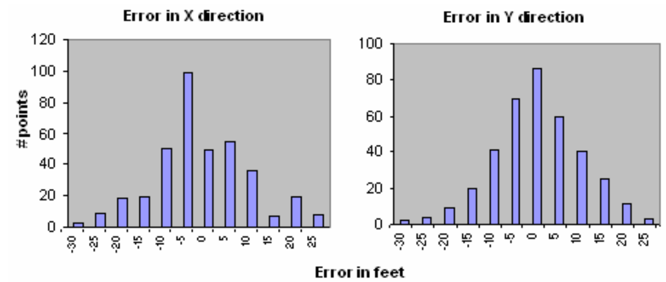


Figure 6. Bias in errors along the x and y directions for k=28.

E. Floor Discrimination

We used our technique described in Section IV.C.1, in conjunction with the data collected from both floors of the two sites, to decide in which floor a given data point belonged. We observed that all data points were correctly classified. We also observed that the decision depth was never more than 3, and in most cases was 1, except for two points in CA. These points had a decision depth of 7 and were very close to an open stairway connecting the two floors. We believe that open stairways will present a challenge, especially in areas like hospitals, hotel lobbies, etc., and additional investigation in these environments could prove interesting.

F. Other Observations

In Section IV.C.2, we described two techniques for matching the client RSS vector against the model. All the experimental results reported in this paper used the Full-NNS technique for matching in the online phase. In our tests, we observed that Full-NNS was consistently better than Top(3)-NNS in our experiments, and sometimes significantly better. We conjecture that the information presented by weak signals or the absence of signals at a location is also significant in location estimation.

We observed in Section IV.A that we need to select a grid size when building the synthetic model. In all our experiments reported in this paper, the grid size used was 3ft × 3ft. We also experimented with a smaller grid size of 1ft ×1ft and did not observe any significant benefits from the reduced grid size. We believe that there are diminishing returns in reducing the grid size below a certain small value. This could be due to the variation of received signal strength at any given location that is caused by shadowing [17], which causes an irreducible error in location estimation.

We observed that the signal strength contours as generated by our technique clearly showed the anisotropy encountered in indoor radio propagation. In particular, as also seen in Figure 7, the waveguide-like effect of corridors was clearly visible.

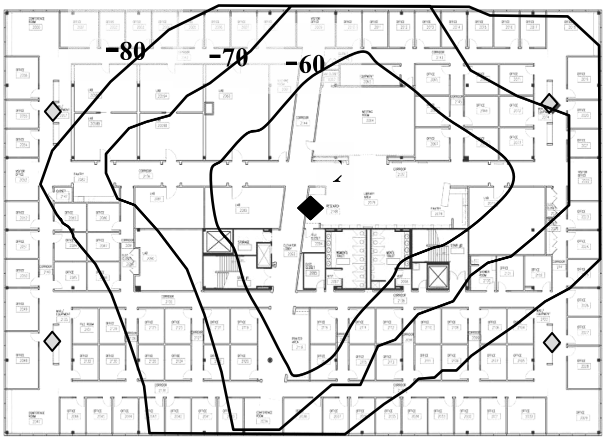


Figure 7. Signal strength contours for the access point at the center of site BR.

The search for the terminal's RSS vector against the synthetic grid may be optimized using commonly known data structure techniques; we omit discussion of this aspect from this manuscript.

VII. COMPARISON WITH OTHER TECHNIQUES

From Section VI it is clear that the performance of the estimation engine in LEASE, despite using a small number of sample points to build its model, is still comparable to other published techniques (mentioned in Section II). The analysis in Section VI does not, however, take into account aspects of the LEASE system such as its adaptability and manageability.

We note that comparing techniques based solely on reported median error metrics might be misleading. Due to vast differences in the environments used for experimentation, the parameters used by various researchers and metrics reported, it is hard to make a direct comparison between techniques. For example, the two largest sites we have seen used in prior work were [4, 25]; in [25], the area of the site is approximately 19000 sq.ft., and in [4] it is about 11000 sq.ft. This compares to our site's area of about 32000 sq. ft. for BR and 34650 sq. ft. for CA. More importantly, all prior work we have seen has data points only in corridors. The open area of corridors is usually more "signal friendly" and no estimates have been provided for the extent of degradation of location estimation when clients were inside offices or labs. (For example, in Section VI.C, we observed that some benefits resulted by only considering corridor data.) The sites profiled in previous work were also "AP-dense" having anywhere from 3—5 APs serving the smaller area. Issues related to not seeing APs at data points were not encountered. Recall that none of the prior work studied the complexity of building and maintaining the model.

In this scenario, we attempt to compare our algorithms with some other published work. It is imperative that we provide a metric for making such comparisons. In subsection A, we propose a metric, and use this metric to compare our approach with [4, 25] in subsection B.

A. A New Metric To Evaluate Location Estimation

A metric for determining the effectiveness of a profiling-based location estimation technique must be able to model at least the following artifacts of the problem: i) the extent of work done in building the model, ii) the location estimation errors seen using the technique, iii) how dynamic the signal environment is at the site, and iv) the adaptability of the technique. We now motivate a metric we call the *effective normalized error,* ε, that takes some of these issues into account.

Intuitively, ε must be directly proportional to some function $f(m)$ of the raw estimation error m (where m can be the mean, median, or similar metric). Let A be the area of a site and let the technique profile the site using k points. Depending on the cost of profiling, the average work done in profiling could be modeled as $g(k)/A$, where $g(k)$ is some function of k. We define the effective normalized error ε as $f(m) \times (g(k)/A) \times h(r,a)$, where $h(r,a)$ is some function of the dynamic nature r of the site, and the adaptability a of the technique. The functions $f(.)$ and $g(.)$ must clearly be monotonic non-decreasing. For simplicity, in the rest of this discussion, we assume $h(r, a) = 1$. However, we note that $h(r,a)$ can take any form; e.g., for sites that have a relatively stable RF environment, $h(r,a)$ could be $1/(g(k)/A)$, thereby negating the impact of the profiling cost in the effective normalized error. Prior work can be considered as modeling $\varepsilon = m$.

If a profiled point "represents" a circular area of radius r around it, such that all clients in that area are mapped to this point, the median error for a uniform distribution of clients within this area is $O(r)$. In this case, $k=1$, and $A = O(r^2)$. This implies that $f(m)$ should logically be $\Omega(r^2)$. The function, $g(k)$, can be any metric that suitably models the cost of profiling.

In this paper, we use the following functional form for ε, namely,

$$\varepsilon(i,j) = \frac{ck^i m^j}{A}, \tag{1}$$

where c is a constant, and $j > 2$. Note that the smaller the ε, the better the technique. The parameters i and j allow us to tune the error metric to emphasize the cost of profiling or the raw error of estimation.

1) Resolving Power

Let $\alpha = A/k$, be the average area per profiled point and let β be the error area (πm^2), where m is the error in location estimation (e.g., the median error). An intuitive measure of how well the location determination works is the dimensionless ratio $\rho = \alpha/\beta$, which we call the *resolving power*. The resolving power indicates how well a technique exploits the available information. All other things being equal, we expect a better technique to have a larger ρ. Note that $\varepsilon(1,3) = c'm/\rho$, where $c' = c/\pi$.

B. Comparison of various techniques using the effective normalized error, ε

Our goal in this section is to compare various techniques with the LEASE estimation engine using the metric from Equation (1) above. We evaluate $\varepsilon(1,3)$ and $\varepsilon(2,3)$ for the various techniques, and use c=1, since we are interested in a comparison only and not the absolute value of ε. Using $\varepsilon(1,3)$

emphasizes reducing the raw error, and using ε*(2,3)* emphasizes reducing the number of profiled points (or number of SEs), *k.* For *m,* we use the median error that is reported by most prior work.

For comparing with other techniques using the metric of effective normalized error, we chose [4], a deterministic technique, and [25], a probabilistic technique, since they have the information necessary to make the required computations. From [4, 25], we extract the required information as follows. For the area profiled, we use the total area of the *corridors* at the respective site, since their experimentation was only done in the corridors. (We assumed a corridor width of 5ft.) We determine the median error, and the number of profiled points from their paper. (We count possible multiple measurements at a profiled point only once.) The raw information we use for calculating ε is presented below in Table 1.

Technique	Area (corridor)	#Training points	Median error
RADAR [4]	≈2920 sq.ft.	70	2.9m (9.6 ft)
Ref. [25]	≈2750 sq.ft.	110	3.5 ft.

Table 1. Comparison of the features of two previously studied approaches.

For the LEASE system for the BR experiments, the area is 32400 sq. ft. For k=12, 28 and 38, the median errors are 15ft, 12ft and 10.4ft., respectively. The computed effective normalized error is shown below in Table 2, where LEASE(*k*) @ BR, represents the LEASE technique with *k* points used to build the model at site BR. Recall that the smaller the ε*,* the better the technique.

Technique	ε*(1,3) (ft)*	ε*(2,3) (ft)*
RADAR [4]	21.2	1484.0
Ref. [25]	1.7	188.6
LEASE(12) @BR	1.2	18.7
LEASE(28) @BR	1.5	41.8
LEASE(38) @BR	1.3	50.1

Table 2. Comparison of effective normalized errors for various techniques.

The lower effective normalized error of the LEASE system shows that it is a very promising approach. It also suggests that our non-parametric modeling technique does a good job of interpolating the signal strength.

VIII. CONCLUSIONS

In this paper, we have studied the problem of location estimation in indoor enterprise wireless networks from the new perspective of easier and automated deployment and maintenance. We have presented a system called LEASE that uses a few stationary emitters and sniffers in a novel way to solve this location estimation problem. Our estimation engine uses non-parametric modeling techniques that automatically capture the anisotropy of received signal strength encountered in indoor environments. Through extensive experiments at two multi-floor sites, we have shown that the location estimation engine in LEASE provides accurate estimates of the floor and location on the floor where the client is located. We have introduced and motivated a new metric called effective normalized error that captures many nuances of the location estimation problem and show that the LEASE system is very effective in terms of this metric when compared to other published techniques.

ACKNOWLEDGMENT

We would like to thank Jim Landwehr, Jack Hong, and Lookman Fazal for many interesting discussions. We would additionally like to thank Jack Hong for help with data collection at the CA site.

REFERENCES

[1] H. Akima, “Algorithm 761: Scattered-Data Surface Fitting that has the Accuracy of Cubic Polynomial,” ACM Transactions on Mathematical Software, Vol.22, No.3, September 1996, pp 362-371.

[2] H. Akima, “Algorithm 760: Rectangular-Grid-Data Surface Fitting that has the Accuracy of Bicubic Polynomial,” ACM Transactions on Mathematical Software, Vol.22, No.3, September 1996, pp 357-361.

[3] H. Akima, “A new method of Interpolation and Smooth Curve Fitting based on Local Procedures,” Journal of the ACM, Vol. 17, No. 4, October 1970, pp 589-602.

[4] P. Bahl, V.N.Padmanabhan, “RADAR: An In-Building RF-based User Location and Tracking System,” Proceedings of IEEE Infocom 2000, Tel Aviv, Israel, March 2000.

[5] P. Bahl, V.N. Padmanabhan, A. Balachandran, “Enhancements to the RADAR User Location and Tracking System,” Microsoft Research Technical Report, February 2000.

[6] R. Battiti, M. Brunato, A. Villani, “Statistical Learning Theory for Location Fingerprinting in Wireless LANs,” Dipartimento di Informatica e Telecomunicazioni, Universita di Trento, Technical Report DIT-020086, October 2002.

[7] T. W. Christ, P.A. Godwin, “A Prison Guard Duress Alarm Location System,” Proc. of the IEEE International Carnahan Conference on Security Technology, October 1993.

[8] S. Ganu, A. S. Krishnakumar, P. Krishnan, “Infrastructure-based location estimation in WLAN networks,” to appear in the Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), March 2004. Also appears as Avaya Labs Technical Report, ALR-2003-053, October 2003.

[9] S. Garg, M. Kappes, “Admission Control for VoIP Traffic in IEEE 802.11 Networks,” in Proceedings of IEEE Globecom 2003, San Francisco, CA.

[10] J. Hightower and G.Borriello, “Location Systems for Ubiquitous Computing,” IEEE Computer, Vol. 34, No.8, pp 57-66, 2001.

[11] G. T. Huang, “Casting the Wireless Sensor Net,” Technology Review, MIT's Magazine of Innovation, July/August 2003, pp 51-56.

[12] Hastie, T and Tibshirani, R. “Generalized Additive Models,” Chapman and Hall, 1990.

[13] J. D. Kraus, “Antennas,” McGraw Hill, 1988.

[14] Andrew M. Ladd, Kostas E. Bekris, Algis Rudys, Guillaume Marceau, Lydia E. Kavraki, Dan S., “Robotics-Based Location Sensing using Wireless Ethernet,” The Eighth ACM International Conference on Mobile Computing and Networking (MOBICOM), Atlanta, GA, September 2002.

[15] P. Prasithsangaree, P. Krishnamurthy, and P. K. Chrysanthis, “On Indoor Position Location With Wireless LANs,” The 13th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2002), Lisbon, Portugal, September 2002.

[16] N.B. Priyantha, A. Chakraborty and H. Balakrishnan, “The Cricket Location Support System,” Proceedings of the Sixth Annual ACM

International Conference on Mobile Computing and Networking, Boston, MA, August 2000.

[17] T. S. Rappaport, "Wireless Communications, Principles and Practice," Prentice Hall, 1996.

[18] T. Roos, P. Myllymäki, and H.Tirri, "A Statistical Modeling Approach to Location Estimation," IEEE Transactions on Mobile Computing, Vol. 1, No. 1, January-March 2002, pp. 59-69.

[19] S. Saha, K. Chaudhuri, D. Sanghi and P. Bhagwat, "Location Determination of a Mobile Device using IEEE 802.11 Access Point Signals," IEEE Wireless Communications and Networking Conference (WCNC) 2003, New Orleans, Louisiana, March 16-20, 2003.

[20] A. Smailagic, D.P. Siewiorek, J. Anhalt, D. Kogan and Y. Wang, "Location Sensing and Privacy in a Context Aware Computing Environment," Pervasive Computing, 2001.

[21] E. Rafajowicz and R. Schwabe, "Equi-distributred Designs in Non-Parametric Regression," Statistica Sinics, vol. 13, pp. 129-142, 2003.

[22] S. Thrun, "Probabilistic Algorithms in Robotics," Technical Report CMU-CS-00-126, Computer Science Department, Carnegie Mellon University, April 2000.

[23] R. Want, A. Hopper, V. Falcao, J. Gibbons, "The Active Badge Location System," ACM Transactions on Information Systems, vol. 10, no. 1, January 1992, pp 91-102.

[24] J. Werb, C. Lanzl, "Designing a Positioning System for Finding Things and People Indoors," IEEE Spectrum, September 1998, pp. 71-78.

[25] Moustafa Youssef, Ashok Agrawala, A. Udaya Shankar, "WLAN Location Determination via Clustering and Probability Distributions," IEEE International Conference on Pervasive Computing and Communications (PerCom) 2003, Fort Worth, Texas, March 23-26, 2003.

[26] Kismet, http://www.kismetwireless.net

[27] Soekris Engineering, http://www.soekris.com

[28] AirDefense, http://www.airdefense.net

[29] Ekahau, http://www.ekahau.com

[30] Newbury Networks, http://www.newburynetworks.com

[31] Wireless Tools for Linux, Open Source Project from HP, http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html

[32] S-Plus, http://www.insightful.com