Physical Security Perimeters for Wireless Local Area Networks

Vishal Bhargava and Mihail L. Sichitiu

(Corresponding author: Vishal Bhargava)

Department of Electrical and Computer Engineering, North Carolina State University
Raleigh, NC - 27695, USA (Email: vishal@iitb.org, mlsichit@ncsu.edu)

(Received Aug. 11, 2005; revised and accepted Oct. 1, 2005)

Abstract

On a wired network, physical authentication is implicitly provided by access: if a user is able to plug a cable into a network socket, he must have cleared other security checks such as the receptionist and/or locked doors. In the case of a wireless local area network (WLAN), the signal propagation is not limited by a fixed boundary, and unauthorized access from outside the security perimeter is possible, and in many instances facile. In this paper, we present a probabilistic technique for localization of users in a WLAN. The presented technique is able to identify intruders based on their location, and thus successfully defend a "parking lot" attack. The approach relies on a probabilistic mapping from received signal strength (RSSI) to location. Calibration inside and around the security perimeter must precede the localization phase. During the localization phase, the RSSI of all the WLAN users is measured by multiple monitoring stations positioned to provide an overlapping coverage of the area (the access points needed to provide the WLAN coverage can double as monitoring stations). A Bayesian technique is used to estimate the location of the unsuspecting mobile user, and the position estimate of each user is updated with every new RSSI measurement at any of the monitoring stations. The presented approach is server-based, i.e., it works without the knowledge or cooperation of the user being tracked, thereby enabling the proposed security application, as well as location-aware services. Validation of the concepts was implemented using an experimental tested in an office environment. The results demonstrate the ability of the proposed technique to estimate the user location to a very high degree of accuracy.

Keywords: Active attacker, localization, physical security perimeter

1 Introduction

Wireless Local Area Networks (WLANs), (especially those compatible with IEEE 802.11 [12]) are fast becoming the networks of choice for enterprises, small offices and households all over the world. With a variety of available inexpensive hardware, WLANs are facing tremendous growth, which is expected to continue in the future. The lack of cables makes WLANs easy to install for system administrators and, at the same time, offer mobility and flexibility for the users. This kind of portability at a reasonable price, without a noticeable drop in bandwidth, has been mainly responsible for WLAN's widespread usage in the home environment.

Unfortunately, with the advent of WLANs, arises the issue of security in a wireless environment. Security in WLANs takes center stage, due to its inherent broadcast mechanism. Every packet that is transmitted can be easily captured by any receiver in its range. Most WLANs do offer some form of security; but as their popularity has increased, a host of flaws have been identified both in the standards, as well as in the implementation of the standards [1, 6, 8, 19].

Intending to correct the flaws discovered in the security of IEEE 802.11, a number of authentication and encryption schemes have been proposed and/or are in the process of standardization: 802.1X, EAP, MAC filtering, 802.11i, TKIP, etc. [5]. A common flaw of all password-based security systems is that the passwords may be poorly chosen or can be obtained through social engineering. Our goal is not to replace those schemes, but rather to *enhance* their effectiveness, by creating a virtual security perimeter inside a physical security perimeter.

The lack of physical boundaries of WLANs creates significant security issues for system administrators. Since the signal range cannot be easily controlled, it is likely that the signal will extend beyond the boundaries created by wired LANs. This leaves the system open to what is commonly known as the parking lot attack (Figure 1). In a parking lot attack, an attacker can eavesdrop on WLAN communication by setting up a laptop with a WLAN adapter in the communication range of the WLAN.

One important feature that is missing from all of the existing proposals for security is the ability to distin-

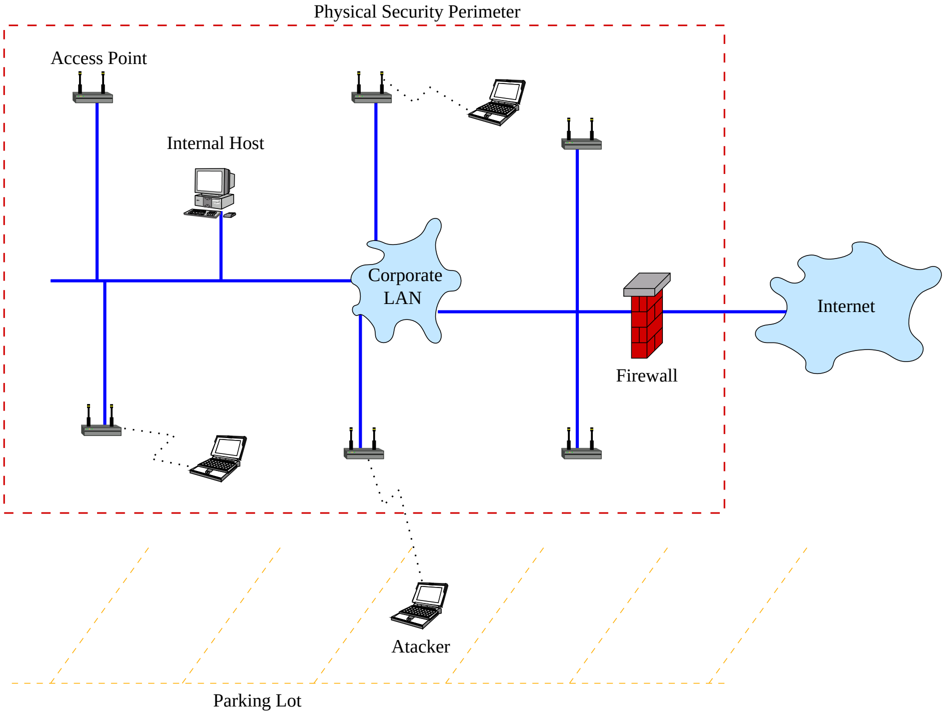

Figure 1: The parking lot attack

guish between the users located within a physical security perimeter and those outside such a perimeter. Such a feature can be used to implementing access restriction based on the physical location of the user. If users outside the security perimeter are not allowed to connect to the organization's WLAN, it would become much more difficult for an attacker to access the network from outside the security perimeter.

In this paper, we propose an algorithm to localize and track a user based on the signal strengths of the packets that he transmits, and thus be able to block the network access of an attacker based on his physical location. The proposed approach aims to restore the properties of the physical security perimeter that was lost with the introduction of wireless networks: only users inside the physical security perimeter would be allowed access, while all other users would be denied access. The solution presented does not require custom equipment; in some cases, only a firmware upgrade of the access points is needed. Alternatively, special monitoring stations might be deployed. The system works without the cooperation (possibly even without the knowledge) of the WLAN users. Since the monitoring stations are passive, it is impossible to detect their presence from the users' point of view.

The emphasis of our work is not on achieving fine grained localization (with precision of a few centimeters), since it is not essential for authentication. The focus of our work is on achieving coarse localization that can answer with high reliability the question, "Is the user inside or outside the security perimeter?" To answer this question we propose a "reverse localization" algorithm that combines Bayesian localization techniques with emergency cellular localization ideas (the base stations collaborate to localize the user, rather than the user actively localizing itself, like in GPS).

The main drawback of the proposed approach is that it can be used to locate and track only an *active* attacker. If the attacker is passive i.e., just eavesdropping, it is impossible to detect him (with this or any other technique). While passive attackers can be extremely dangerous by gathering sensitive information, arguably, the active attackers can cause the maximum amount of damage. Furthermore, the approach is only applicable if the coverage area is protected by a physical security perimeter (e.g., a building or a military base), thus being not suitable for public access WLANs (e.g., airports, hot-spots, etc.).

The paper is organized as follows: Section 2 discusses the related work in the area of localization. Section 3 presents our proposed localization algorithm. Thereafter, we present the results of our tracking and localization experiments in Section 4. Section 5 concludes the paper.

2 Related Work

The localization field is a rather mature field, with significant research activity in many application areas The problem is known in literature under many names, including localization, locationing, geolocation, positioning, etc. An excellent survey of the area was published a couple of years ago [10].

The Global Positioning System (GPS) is perhaps the most well-known positioning system currently in use.

The US Federal Communications Commission's E911

Table 1: Classification of RF-based localization techniques

	Indoor	Outdoor
Host-based	RADAR, etc	GPS
Server-based	*Proposed technique*	E911 services

telecommunication initiatives require that wireless phone providers develop a way to locate any phone that makes a 911 emergency call. Significant research was thus geared towards localizing the cellular phone users in outdoor environments. Many applications call for indoor solutions to the problem of localization. Nowadays, many companies offer a variety of solutions based on visual tracking, ultrasound, or even radios with dedicated hardware [7, 11, 14, 17, 18, 20, 21]. The popularity of WLANs, (especially of the IEEE 802.11 standard), sparked a significant interest in indoor localization systems based on the already available 802.11 access points (used for radio coverage of a larger WLAN) [3, 4, 9, 15, 16].

According to [10], the localization systems can be classified in *host-based* and *server-based* systems. In a host-based system, the users gather information from the infrastructure with the goal of localizing themselves; a classical example is the GPS system. In a server-based system, the infrastructure gathers information from the users and determines the location of the users; such systems are presented, for example, in [3, 4, 9, 15, 16]. Similarly, the RSSI-based localization systems can be classified in outdoor and indoor systems. The main difference between the two types of systems is that outdoors, many times, the assumption of a circular propagation pattern holds. In this paper we are presenting a *server-based, indoor localization system* (see Table 1), and evaluate its suitability for a physical authentication system. The localization system can have many other applications, e.g., tracking of inventory items, personnel, location aware services, etc.

3 Proposed Approach

In the proposed approach we use a network of monitoring stations spread over the coverage area of the WLAN. Each monitoring station listens on the wireless medium, and captures all packets that are correctly received (they operate in "promiscuous" mode, i.e., it does not filter the received packets by its own MAC address). Upon the receipt of a packet, the monitoring station also measures the RSSI associated with that packet, and it then sends the MAC address of the sender and the RSSI measurement to a central server that combines the information from all monitoring stations into a best estimate of the position of all users (both authorized users and attackers). A system administrator can thus create policies denying WLAN access to users outside certain areas (see Figure 2).

The proposed localization and tracking technique can be divided into two phases:

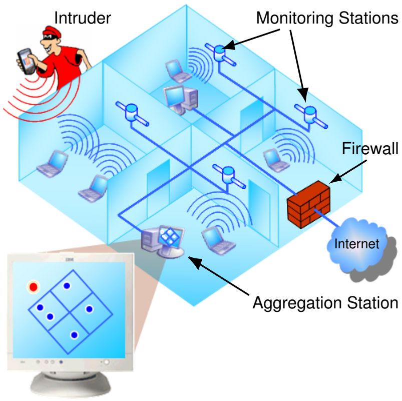

Figure 2: Intrusion detection based on user localization

- The *calibration* (sometimes referred to as the "finger-printing") phase; the purpose of the calibration phase is to collect labeled data (packets labeled with the location of the sender) from different points spread over the tested, and then filter this data into a usable form for the localization phase.
- The *localization* phase; the goal of the localization phase is to estimate the location of the source(s) of unlabeled data (i.e., any 802.11 packet). Somewhat similar to the approach presented in [15], we follow a Bayesian approach for dynamic state estimation. The state of the system, at any given time, is represented by a probability distribution function defined as the probability of a sender being present in a given area. The state is updated as a function of the previous state and new data measurements. We have used a recursive filtering approach, in which each new measurement is processed individually, and the received data is processed sequentially rather than as a batch.

3.1 Calibration

The calibration (or finger-printing) phase is a key step in which the system "learns" the mapping of the probability density function as a function of the RSSI. In this phase, labeled data (location of the sender of a packet and the received signal strength) is collected, and processed into a simplified form that will later be used by the localization phase to estimate the position of the wireless user in real-time.

In practical terms, in the calibration phase, a mobile node periodically broadcasts its position at different positions in the tested; each of the monitoring stations captures these packets along with the RSSI corresponding to the packet. This information is further sent to the server from each of the monitoring stations. Corresponding to each packet received by a monitoring station, a tuple consisting of the location (x,y coordinates), RSSI and the monitoring station identifier ($MSID$) of the MS that captures the packet is sent to the server. An almost equal number of labeled packets with varying RSSI is expected at each of the monitoring stations, if they are spread uniformly over the tested and have an almost equal range. The packets are collected both from "inside" as well from "outside" the physical security perimeter.

For simplicity, assume that the tested is divided into an $N \times N$ grid represented as a matrix, each element of a matrix corresponding to a square of the surface of the tested. After the calibration phase the server will have a database with one entry for each packet received by a monitoring station, each entry being of the form $(< x, y >, MSID, RSSI)$, where $< x, y >$ is the location of the mobile station at the time when the packet was sent, MSID is the identifier of the monitoring station that received the packet (more than one monitoring station may receive a packet), and RSSI is the received signal strength of the received packet received.

For a collection of data elements with a constant MSID=MS_i and a constant RSSI=λ, let $\eta_{i,j}$ be the number of elements at location (i, j) on the X-Y plane. In this case, corresponding to a packet received by monitoring station MS_i with a signal strength λ, the probability that the user is at position (x, y) is:

$$p_{i,\lambda}(x,y) = \frac{\eta_{x,y}}{\sum_{i=1}^{i=N}\sum_{j=1}^{j=N}\eta_{i,j}}. \qquad (1)$$

During the localization phase, these measurements become constraints on the position estimates of the users. Thus, the constraint $c(n)$ resulting from capturing the n^{th} packet is:

$$c(n) = \begin{pmatrix} p(x_1,y_1) & p(x_2,y_1) & \dots & p(x_N,y_1) \\ p(x_1,y_2) & p(x_2,y_2) & \dots & p(x_N,y_2) \\ \vdots & \vdots & \ddots & \vdots \\ p(x_1,y_N) & p(x_2,y_N) & \dots & p(x_N,y_N) \end{pmatrix},$$

where $p(x, y)$ is given by Equation (1).

It would be very tedious and error prone to send calibration data from *each* square of the $N \times N$ matrix. Instead we collect data from a relatively small, approximately uniformly distributed, points and interpolate the results using a Gaussian filter [13] similar to the one described in Section 3.2.1.

3.2 Localization

In the localization phase, each monitoring station upon the reception of a packet will send at the server a triplet with the MAC address of the sender of the received packet (readily available in the 802.11 MAC header), the signal strength of the packet and the monitoring station identifier. The goal of the server is to determine the probability distribution function for each user by combining the data collected from multiple monitoring stations.

We define the position estimate of a user $z(n)$, after the server received the n^{th} measurement from that user as the probability density function (pdf) of the position on the X-Y plane. We chose to represent this probability as a two-dimensional matrix:

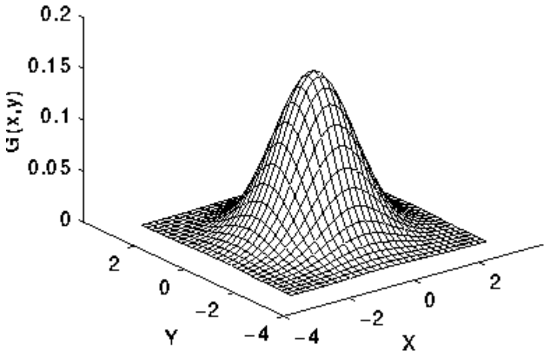

Figure 3: 2-D Gaussian distribution with mean (0,0) and $\sigma = 1$

$$z(n) = \begin{pmatrix} p(x_1,y_1) & p(x_2,y_1) & \dots & p(x_N,y_1) \\ p(x_1,y_2) & p(x_2,y_2) & \dots & p(x_N,y_2) \\ \vdots & \vdots & \ddots & \vdots \\ p(x_1,y_N) & p(x_2,y_N) & \dots & p(x_N,y_N) \end{pmatrix},$$

where $p(x_i, y_i)$ is the probability of finding the user at location (x_i, y_i).

For localization, we define a model that relates this state with the corresponding tuples (MSID, RSSI) received from the monitoring stations. We also consider the change in state over time, due to movement of the mobile node. Any algorithm concerning a dynamic system consists of at least two models [2]:

- *System model* - a model describing the evolution of the state with time;
- *Measurement model* - a model relating each measurement with the change in state.

3.2.1 System Model

We assume that the mobile nodes that we aim to track are mobile, and that their random movement has a normal distribution, centered at the starting location. Thus, we periodically perform the convolution of the state with a Gaussian filter. The effect of this convolution operation is the "spreading" of the position estimate of a user reflecting the increase in the uncertainty due to user movement. In 2-dimension, an isotropic Gaussian (see Figure 3) is of the form

$$G(x,y) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

Thus, the state is periodically updated:

$$z(n+1) = z(n) * G, \qquad (2)$$

where, by $*$ we denote the convolution function.

3.2.2 Measurement Model

The measurement model defines the relationship of the state estimate to successive measurements. Section 3.1 describes the algorithm to compute the constraints over the X-Y plane for a given pair of RSSI and MSID values. The computed 2-dimensional constraint is used to update the position estimate of the node:

$$z(n+1) = z(n) \cap c(n), \tag{3}$$

where $z(n+1)$ is the new position estimate, $z(n)$ is the old position estimate, and $c(n)$ is constraint resulting from the measurement. The position estimate is initialized with a uniform distribution:

$$z(0) = \frac{1}{N^2}\begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}. \tag{4}$$

An important factor that has to be considered while developing an algorithm is the reproducibility of the experimental data. With 802.11 wireless cards (as with practically all wireless transceivers) there is a significant variance in the RSSI over time. Thus, for every received signal strength λ we average all constraints resulting from signal strengths between $\lambda - \beta$ and $\lambda + \beta$.

Summary

The localization algorithm, including the system model and the measurement model, can be summarized as:

1) Initialize the position estimate ($z(0)$) as the entire space (as in Equation (4)).

2) For each packet received from a monitoring station:

   - if the packet received from monitoring station MS_i with RSS λ_i, calculate the average constraint $c(n)$ by averaging over the constraints in the range $[\lambda_i - \sigma \dots \lambda_i + \sigma]$; and
   - update the position estimate $z(n+1)$ according to Equation (3);

3) Periodically, update the system model as Equation (2).

4) Goto Step 2.

Once the localization procedure results in position estimates, is relatively easy to exclude (either at the access points, or at the common switch/router) all users local outside the security perimeter.

A denial of service attack that uses the proposed system can be imagined. An attacker can capture the MAC address of a legitimate user and transmit packets from outside the security perimeter. The system will then localize the attacker and deny access both to the attacker and the legitimate user. However, this is arguably better than allowing both of them access to the network.

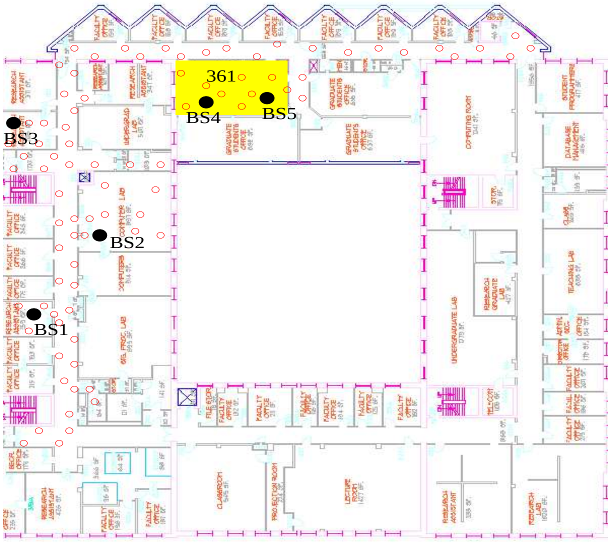

Figure 4: The experimental tested

4 Experimental Setup and Results

Initially we planned to use a network simulator to evaluate the performance of such a system. However, all current network simulators have inadequate physical layer models for indoor environments. Hence, in order to evaluate the performance of the system, we decided to implement a system prototype.

4.1 Experimental Setup

The experimental setup consists of five monitoring stations (MS), a mobile node (MN), and a server. The MN is connected at the campus WLAN, and any packet being transmitted by the mobile node is received by the MSs. Each MS has a wireless as well as a wired interface. It captures all packets on the wireless interface and transmits the RSSI, Source MAC and MSID tuple to the server, using the wired network. The server which is connected to the monitoring stations via the wired network, receives and processes the information to estimate the location of the MNs.

The experimental tested is located on the third floor of the Electrical and Computer Engineering Department in Daniels Hall. We have limited the experimental area to two perpendicular corridors of the Department (Figure 4). We set up a network of 5 monitoring stations relatively uniformly distributed over the tested. The monitoring stations are placed in such a way that their range covers the entire tested. Our aim is to track WLAN users anywhere along the two corridors, and in the adjoining rooms. The yellow area, Room No. 361, acts as the experimental "outside the perimeter" for the purpose of our tested. Figure 4 shows a floor map of our tested. The map in Figure 4 is divided into a grid 100 × 100; whereas the dimensions in reality are $66m \times 73m$. Each cell is thus 0.66m on the x-axis and 0.73m on the y-axis. The position of the calibration readings is shown with small red circles in Figure 4.

- Monitoring Station: The setup consists of an iPAQ (Model No. H3870) connected to a computer, via a serial link supporting data transfer rates of up to 115 Kbps. Each iPAQ is equipped with an Orinoco gold Classic PC card for capturing the packets of the tracked users. The iPAQs run Familiar Linux 0.6 (Opie) with kernel 2.4.18-rmk3. Each of the computers attached to the iPAQ has a 333 MHz Pentium II processor, running RedHat Linux 7.3 (kernel 2.4.18-3).
- Server: Our server is a desktop with a 2.4 GHz Intel Pentium 4 processor, 512 MB RAM, running RedHat Linux 7.3 (kernel 2.4.18-3).
- Mobile Node: A laptop with a 1.8 GHz Intel Pentium 4m processor, 256 MB RAM, running RedHat Linux 8.0 (kernel 2.4.18-14), acts as the mobile node. A Cisco Aironet 350 series PC card is used as the transmitter (the Cisco cards are capable of varying their transmission power).

The goal of our experiment is to locate an intruder and identify whether he is inside or outside a given security perimeter with a very high degree of reliability. For our implementation, we set the security perimeter as the boundary of Room No. 361 (Figure 4). A user inside the room is considered an intruder. In order to *compare* the estimated position of the user with his real coordinates, the real position of the user has to be known. A utility enabling the user to specify his real position by clicking on the map, was developed. This position was considered as the "real position" of the user and compared with the estimated position, which is calculated using the measured RSSI of each of the packets transmitted by the user.

The following metrics are used to judge the efficacy of our implementation.

- *Error of location estimate* defined as the Euclidean distance between the real position of the user and the estimated position given by our implementation. The performance of the algorithm with respect to the error in estimation can be visualized by plotting the percentile of the error estimate against the error in distance. Another measure that reflects the performance is the average error in estimating the user's location.
- *Misinterpretation of position:* A misinterpretation occurs when the actual position of the user is different from the estimated position, with respect to the security perimeter. Misinterpretation of position can be of two types (a) False Positives and (b) False Negatives.
  - A *False Positive* is the case when the estimated position of the user is outside the security perimeter (positive), although the real position of the user is inside.
  - A *False Negative* is the case when the estimated position of the user is inside the security perimeter (negative), although the user is actually outside the security perimeter.

  The probability of False Negatives is a very important metric, as the objective is to detect the intruder effectively (a user outside the security perimeter). It is more important to be able to detect and block intruders without any exceptions than to allow a legitimate user close to the security perimeter (False Positives). The probability of False Negatives should ideally be zero. The probability of False Positives is also an important measure which would reflect the inconvenience caused to a legitimate user in the vicinity of the security perimeter. It is a measure of the probability that if a user has been detected outside the perimeter, what is the chance that he is actually a legitimate user inside the security perimeter.

During the implementation, a number of parameters had to be chosen. These design parameters can be divided into two sets corresponding to the calibration and localization processes.

One of the primary calibration parameters is σ_{loc}, which is the standard deviation for the Gaussian filter used to interpolate between the data-point measurements. During the data collection process, selected data points spread uniformly over the tested were chosen for the sake of convenience and to reduce the time taken to calibrate the tested. A Gaussian filter was then applied over the collected data, to generate new data points over the entire tested. Some of the factors to be considered while choosing σ_{loc} were:

- The indoor signal strength does not have a normal distribution, as indoor obstructions (e.g., walls) cause a significant attenuation in signal power. Thus, σ_{loc} should be sufficiently small (< 0.5m) considering the fact that actual RSSI measurements can vary significantly across very short distances (e.g across a wall).
- RSSI variation is very small over short open spaces (e.g open corridors), and we did not observe a noticeable change in measured RSSI between locations 2-3 meters apart. This would lead us to set σ_{loc} to as much as 2-3m.

- In our implementation, our cell size is about 0.65m on the X-axis and 0.70m on the Y-axis. When we apply the Gaussian filter, we scale the Y-axis correspondingly to make the filter circular in nature, and thus keep a unit on the Y-axis equivalent to the unit on the X-axis, which is 0.65m.

Considering these factors, we choose the value of $\sigma_{loc} =$ 1m which is a compromise solution considering the above factors. Thus, $\sigma_{loc} \approx 1.5$ cell units.

The system model for the localization process was discussed in Section 3.2.1. The system model helps in continuous tracking of the mobile user. The main parameter in this model is σ_{sys}, which is the standard deviation of the Gaussian filter that is applied to the probability distribution matrix of the user at periodic (one second in our case) intervals. The Gaussian filter is applied to account for the movement of the user and to track the user. Thus, assuming that with a 97% probability that the mobile user does not move at a speed of more than 3m/s, we keep $3\sigma_{sys} =$ 3m, therefore $\sigma_{sys} =$ 1m. Thus, $\sigma_{sys} \approx 1.5$ units.

The measurement model (Section 3.2.2) describes our approach at finding the constraint matrix to update the estimated position every time a new packet is received. The constraint matrix is calculated by averaging probability density matrices for a range of RSSI values over which we can assume the RSSI to vary for the same location. During calibration, a large number of packets were collected for every position rotating the laptop around the point to account for variation in RSSI due to the directionality of the wireless adapter. This data collection method causes a variation in the RSSI, but it might not be sufficient to reflect the variation in RSSI over time. We noticed that the range of values we collected from the MS show a maximum variation in RSSI, (β) of about three units. Thus, to produce a constraint we will average the constraints from $2\beta = 6$ different RSSI values, (similar to the deviation in RSSI assumed in [4]).

Thus, the main three parameters that may affect the results of the proposed algorithm are chosen as:

$$\begin{aligned}\sigma_{loc} &= 1.5,\\ \sigma_{sys} &= 1.5,\\ \beta &= 3,\end{aligned}$$

where the units are a function of the cell size.

4.2 Experimental Results

Figure 5 shows the cumulative probability of the localization error, i.e., on the y axis there is the probability that the localization error is smaller than the value on the x axis. There is a 30% chance that the estimated location is less than $0.95m$ from the real position, at least 50% of the errors are less than $1.5m$, and 90% less than $3.3m$. The average estimation error is $1.73m$. Other metrics to observe are:

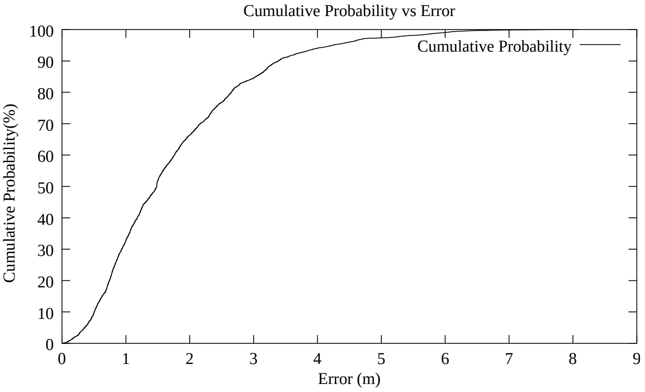

Figure 5: Cumulative Probability of Error vs. Error in estimation $\sigma_{loc} = 1.5, \sigma_{sys} = 1.5$ and $\beta = 3$

- False Negatives = 0. This means a 0% error in reliably estimating if the user is outside the security perimeter. Thus, we are reliably able to determine if the user is outside the security perimeter.
- False Positives = 10.4%. This translates to an approximately 10% chance that the user, who has been estimated to be outside the security perimeter, is actually inside the perimeter. When the user is close to the edge of the security perimeter, or standing close to an exit, he could be mistakenly identified as being outside the perimeter. This relatively high number of False Positives is mainly due to the inaccuracies in measuring the real position of the user during the measurements.

4.3 The Effect of Variation in Design Parameters

In this section, we evaluate the effect of the design parameters on the metrics of interest (localization error, False Positives and False Negatives). We will change one parameter at a time, while keeping the other parameters constant.

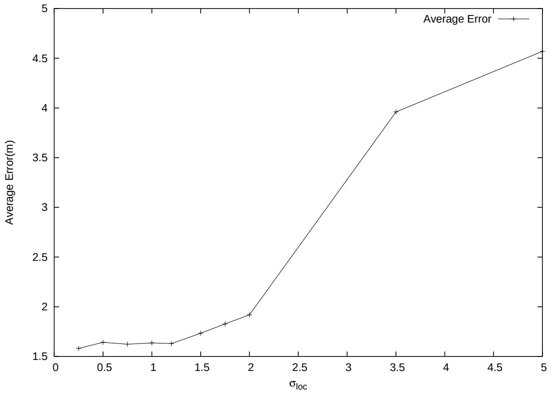

Figure 6: Variation of average localization error vs variation in standard deviation of the Gaussian filter used for calibration (σ_{loc}), with $\sigma_{sys} = 1.5$ and $\beta = 3$

4.3.1 The Effect of the Variation of the Standard Deviation of the Gaussian Filter used for Calibration (σ_{loc})

Figure 6 shows that the average error in estimation increases with an increase in the standard deviation (σ_{loc}) for the Gaussian filter used in the calibration process. The average error does not increase with the decrease of σ_{loc} due to the relatively dense data points collected during the calibration phase (i.e., interpolation is not really needed). On the other hand, when σ_{loc} increases, we effectively reduce the resolution of the system during the calibration phase. The possibility of a wrong estimate increases as multiple locations might have a similar set of RSSI values.

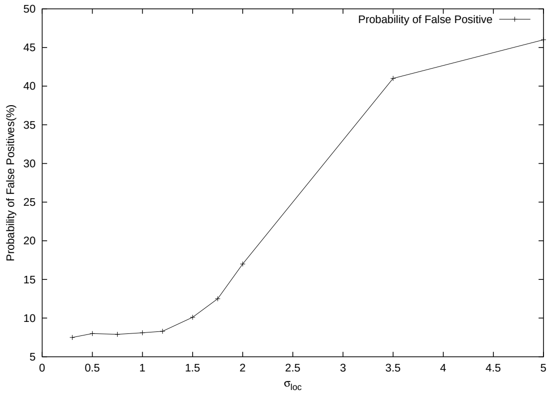

Figure 7: The effect of varying the standard deviation of the Gaussian filter used in calibration (σ_{loc}) on probability of false positives, keeping $\sigma_{sys} = 1.5$ and $\beta = 3$

In Figure 7, we observe that the False Positives increase with an increase in the standard deviation of the Gaussian filter used for σ_{loc}. Figure 7 closely correlates well with Figure 6, (the average error). As the error of estimation increases, we see an increase in the number of False Positives.

An interesting observation is that the probability of False Negatives remains zero when varying σ_{loc}. This could be due to the large difference in measured RSSI when the user is outside the security perimeter(i.e in room No.361 for our implementation) and when it is inside, because of the presence of two monitoring stations in Room 361. None of the points outside Room 361 (i.e., inside the security perimeter) would provide an alternate solution for the set of RSSIs because of the large difference in RSSI inside and outside the room. Also, the effect on the estimated position is minimal because we use a circular Gaussian filter which spreads the values circularly over the region, and the mean remains the same. If the probability density is very high inside a particular area, the estimated position would remain in the area even on applying a Gaussian filter with a high standard deviation.

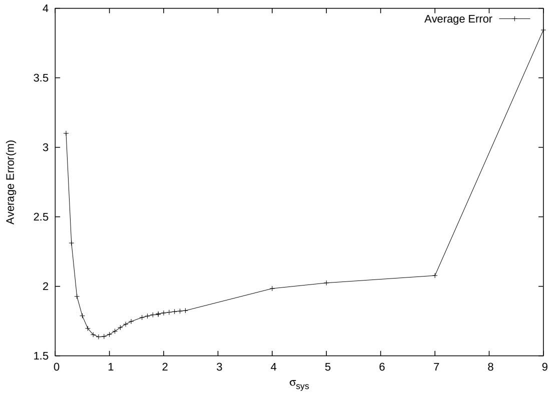

Figure 8: The effect of varying the standard deviation of the Gaussian filter in the system model(σ_{sys}) on the average localization error, with $\sigma_{loc} = 1.5$ and $\beta = 3$

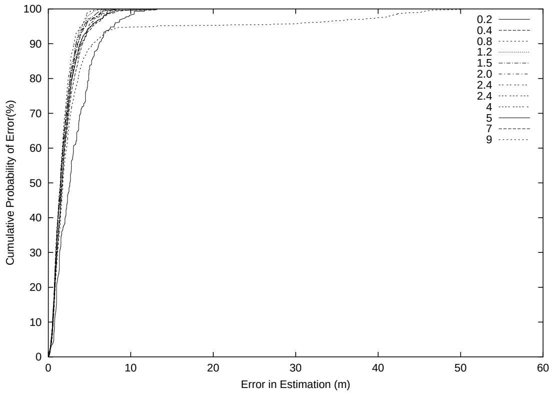

Figure 9: Effect of varying σ_{sys} on the Error distribution, $\sigma_{loc} = 1.5$ and $\beta = 3$

4.3.2 The Effect of Varying the Standard Deviation used for Gaussian Filtering in the System Model (σ_{sys})

In Figure 8, we observe that the average error in estimation has a minimum for $\sigma_{sys} \approx 0.8$. The value of the minimum is dependent on the average speed of the user during the test-run. While assuming $\sigma_{sys} = 1.5$ in Section 4.1, we assumed the maximum speed of a user to be 3 m/s for calculating a σ_{sys}, which would lead to a stable system enabling continuous tracking of the user. While collecting data for the experiment, the average speed was much lower. In this case, even though the minimum is clearly around $\sigma_{sys} = 0.8$, it is better to keep σ_{sys} high to maintain a stable system that can track mobile users moving at a higher speeds.

In Figure 10, it can be seen that the probability of False Positives increases with the increase in the standard deviation of the Gaussian filter used in the system model. The probability of False Positives first decreases on varying σ_{sys} from 0.2 to 0.3 and increases for values of $\sigma_{sys} > 1$. The decrease for values between 0.2 and 0.3 can be attributed to the difference in the tracking speed with respect to the actual average speed of the user during the course of the experimentation. The increase in

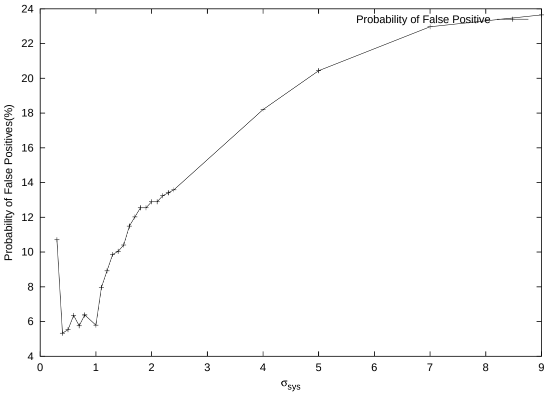

Figure 10: The effect of varying the standard deviation of the Gaussian filter used in the system model (σ_{sys}) on the probability of False Positives, with $\sigma_{loc} = 1.5$ and $\beta = 3$

the probability of the False Positives with an increase in σ_{sys} can be attributed to the spreading of the probability densities by the Gaussian filter. Even a slight change in the estimated position can cause a sharp increase in the number of False Positives due to the rigid boundary.

Our model performs very well in deciding that users that are actually outside are outside the security perimeter. The probability of False Negatives is always less than a 1%; in fact it is exactly equal to 0%, for $0.4 < \sigma_{sys} < 1.9$. Thus, the system is able to locate the intruder that is outside the security perimeter with 100% accuracy.

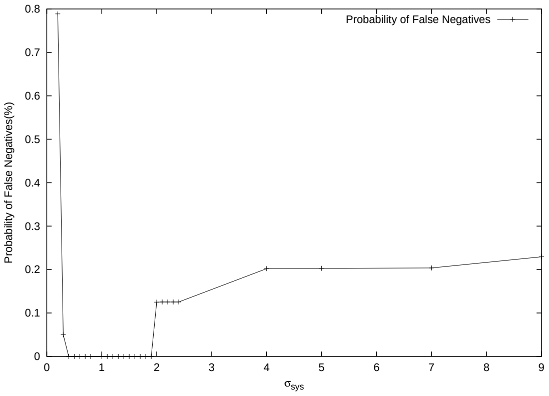

Figure 11: Effect of varying the standard deviation of the Gaussian filter used in the system model (σ_{sys}) on the probability of False Negatives, with $\sigma_{loc} = 1.5$ and $\beta = 3$

4.3.3 The Effect of Varying the Range of the measured Received Signal Strength (β)

Figure 12 shows that by varying the assumed variance in the RSS (β), the average error starts increasing after $\beta > 6$. In this case the resolution of the RSSI measurements is reduced and thus it is expected that the precision of the localization will suffer. For $1 < \beta < 6$, the average error in estimate is almost constant. This means that for a specified location, the received signal strength at a particular MS does not vary by more than 6 RSSI units.

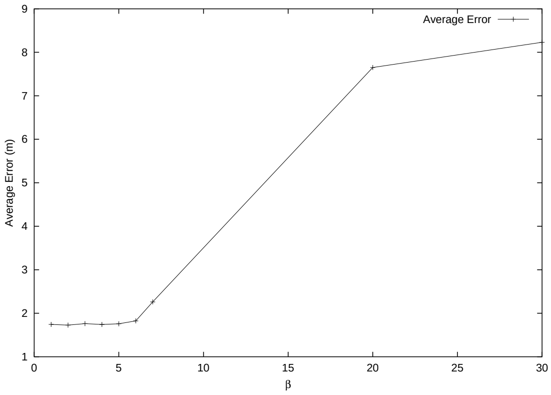

Figure 12: The effect of varying the assumed variance in RSS(β) on the Average Error, with $\sigma_{loc} = 1.5$ and $\sigma_{sys} = 1.5$

The average localization error is just as small for $\beta = 1$ as for higher values of β. This somewhat unexpected result can be explained by the fact that during the data collection process, we obtained data for calibration by sending a large number of packets transmitted in different directions. The natural variation in the RSSI of these packets might be satisfying almost the entire set of RSSI values that can be recorded at a MS from a particular position. However, if we consider a zero deviation in RSSI, i.e., for $\beta = 0$, the system is not able to locate the user and becomes unstable.

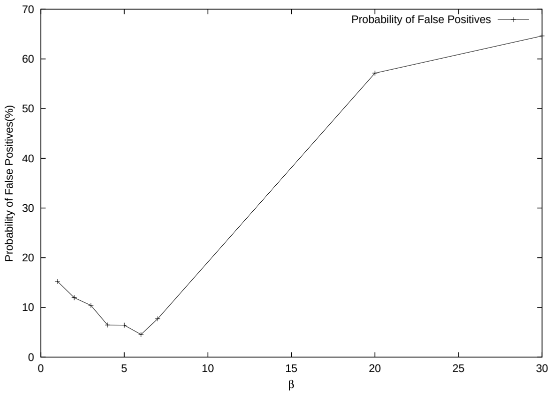

Figure 13: Effect of β on False Positives, $\sigma_{loc} = 1.5$ and $\sigma_{sys} = 1.5$

Figure 13 shows the effect of varying β on the probability of False Positives. We observe that even though the average error in estimate hardly varies for $1 < \beta < 6$, there is a drop in the number of False Positives for the same region. This justifies our assumption of a variation in observed RSSI. The number of False Positives rises dramatically as we increase β and it also corresponds with an increase in the average estimated error.

The probability of a False Negative is zero for all values of $2 < \beta < 30$. At $\beta = 1$, there exists a small probability of 0.2%, of estimating the user inside the security perimeter even though he is actually outside.

4.3.4 The Effect of Varying the Transmission Power

If the system is calibrated at a certain transmission power level, and the intruder accesses the network while using a different transmission power level, the system may not be able to localize it exactly. We decided to estimate the localization error as a function of the transmission power of the WLAN users. Thus, we performed the calibration at a power level of 20mW, and then varied the transmission power levels of the mobile node from 1mW to 100mW and observed the localization error.

The Cisco Aironet wireless adapter allowed us to vary the transmission power from 1mW to 100mW. *All* existing WLAN adapters use a transmission power in this range (using a higher power is basically useless as the access points themselves use up to 100mW transmitters). Thus, in addition to robustness to power transmission level, this experiment reflect the robustness to different adapters (we also verified with a Lucent Orinoco card). We observed a variation of about 10-12 units of RSSI between power levels of 1 mW and 100mW. This deviation is relatively small when compared to the deviation due to the change in location.

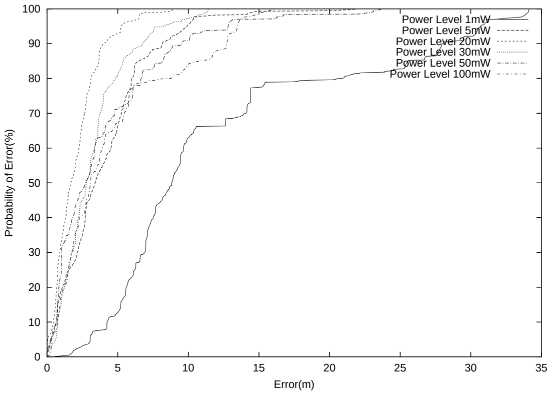

Figure 14: The effect of varying transmission power on the cumulative probability of the localization error, with $\sigma loc = 1.5$, $\sigma_{sys} = 3$ and $\beta = 7$

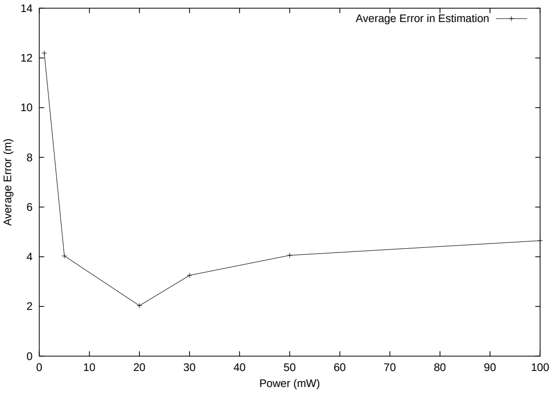

Figure 15: Average Error vs Transmission Power, with $\sigma loc = 1.5$, $\sigma_{sys} = 3$ and $\beta = 7$

In Figure 14 and Figure 15, we see that the average error in estimation is the smallest for 20mW (the power used during calibration). This is as expected. We had to keep the $\beta = 7$ for the estimation to converge in the case of 1mW and 100mW runs.

For a transmission power level of 1mW, the average error in position estimate is about 11.88m; it decreases to 3.96m for 5mW and 1.8m for 20mW, and increases back to 4.2m at 100mW.

The probability of a False Positive for 20mW is about 10%, which means that 10% of the estimated positions outside the security perimeter are actually inside the perimeter. This number increases to 60% at a power level of 50mW.

The probability of a False Negative for 5mW is around 5%, which is still low, but not 100% accurate. At power levels of 20mW, 30mW and 50mW, the probability of a False Negative is 0%, thus indicating that the intruder is always detected.

Thus, the change in the power level at the transmitter is almost irrelevant for the accuracy of the proposed approach. Essentially the path loss significantly out-weights the change in transmission power, and an attacker could not defeat the proposed approach by changing (even often) its transmission power.

4.3.5 Fault Tolerance

To evaluate the effect of the failure of the monitoring stations, we switch off two monitoring stations, one at a time. The MSs are shown in Figure 4. We consider two different cases (a) MS_3 failed and (b) MS_3 and MS_4 failed.

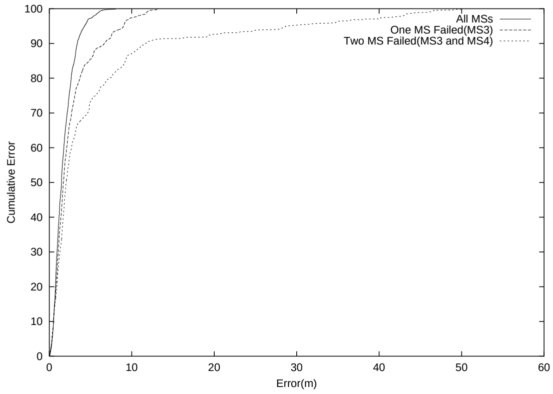

Figure 16: The effect on performance of the localization system with the failure of MSs, with $\sigma loc = 1.5$, $\sigma_{sys} = 3$ and $\beta = 3$

Figure 16 shows that the localization error increases if there is a fault in the monitoring network. However, the localization process does not fail. Even when two out of the five MSs fail, we are able to locate the user with considerable accuracy.

- MS_3 failed:
  Average Error = 2.56m, False Negatives = 0.16%, False Positives = 10.8%.

- MS_3 and MS_4 failed:
  Average Error = 5.52m, False Negatives = 0.25% and False Positives = 25.55%.

If more than two MSs fail, the system becomes unstable and only users in a small part of the tested are successfully located.

5 Conclusion

We proposed and implemented a novel server-based approach to locate the users of a WLAN using only the received signal strength of packets transmitted by the WLAN users. Even though our initial intention was to provide only a coarse-grained localization, and an extremely reliable method to determine if a user is outside the defined security perimeter, our implementation was able to locate and continuously track users with an average error in estimated position of around $1.65m \approx 5.4$ feet, which is rather good for indoor localization in a WLAN. We were also able to achieve an almost 100% accuracy in identifying the intruder (user outside a fixed security perimeter). The learning process in our approach, i.e., the calibration phase, took about 30 minutes for our tested. Thus, this technique offers a very low lead time for deployment of a new setup. We did not use any specialized hardware for the implementation, although we did use extra monitoring stations to detect the RSSI of the user's packets. The ideal implementation would be a small hardware upgrade on the access points which would enable them to double up as monitoring stations.

References

[1] W. A. Arbaugh, N. Shankar, and Y. J. Wan, "Your 802.11 wireless network has no clothes," *IEEE Wireless Communications*, pp. 44V51, Dec. 2002.

[2] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, Feb. 2002.

[3] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceeding of Infocom'2000*, vol. 2, pp. 775-584, Tel Aviv, Israel, Mar. 2000.

[4] P. Bahl and V. N. Padmanabhan, *Enhancements to the Radar User Location and Tracking System*, Microsoft Research Technical Report MSR-TR-2000-12, 2000.

[5] L. Barken, *How Secure is Your Wireless Network?*, Prentice Hall, 2003.

[6] N. Borisov, I. Goldberg, and D. Wagner, *Intercepting Mobile Communications: The Insecurity of 802.11*, http://www.isaac.cs. berkeley.edu/isaac/wep-faq.html, 2001.

[7] M. H. T. Darrell, G. Gordon and J. Woodfill, "Integrated person tracking using stereo, color, and pattern detection," in *Proceeding of Conf. Computer Vision and Pattern Recognition*, pp. 601-608, Los Alamitos, CA, 1998.

[8] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," LNCS 2259, pp. 1V20, 2001.

[9] M. Helén, J. Latvala, H. Ikonen, and J. Niittylahti, "Using calibration in RSSI-based location tracking system," in *Proceeding of the 5th World Multiconference on Circuits, Systems, Communications & Computers (CSCC20001)*, 2001.

[10] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, no. 8, pp. 57-66, Aug. 2001.

[11] J. Hightower, R. Want, and G. Borriello, *SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength*, Technical Report CSE 2000-02-02, University of Washington, Seattle, WA, Feb. 2000.

[12] IEEE, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, June 1999.

[13] K. Ito and K. Xiong, " Gaussian filters for nonlinear filtering problems," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, May 2000.

[14] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for easy living," in *Proceeding of 3rd IEEE Intl. Workshop on Visual Surveillance*, pp. 3-10, 2000.

[15] A. M. Ladd, K. E. Bekris, G. Marceau, A. Rudys, L. E. Kavraki, and D. Wallach, "Robotics-based location sensing using wireless ethernet," in *Proceeding of Eighth ACM International Conference on Mobile Computing and Networking (MOBICOM 2002)*, pp. 227-238, Atlanta, Georgia, Sept. 2002.

[16] J. Latvala, J. Syrjärinne, H. Ikonen, and J. Niittylahti, "Evaluation of RSSI-based human tracking," in *European Signal Processing Conference*, pp. 2273-2277, 2000.

[17] R. Orr and G. Abowd, "The smart floor: A mechanism for natural user identification and tracking," in *Proceeding of the 2000 Conf. Human Factors in Computing Systems (CHI 2000)*, pp. 275-276, 2000.

[18] K. Pahlavan, X. Li, and J. Makela, "Indoor geolocation science and technology," *IEEE Comm Soc. Mag.*, pp. 112-118, Feb. 2002.

[19] A. Stubblefield, J. Ioannidis, and A. Rubin, *Using the Fluhrer, Mantin, and Shamir Attack to Break WEP*, ATT Labs Technical Report TD-4ZCPZZ, Revision 2, Aug. 2001.

[20] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office," *IEEE Personal Communications*, vol. 4, no. 5, pp. 42-47, Oct. 1997.

[21] J. Werb and C. Lanzl, "Designing a positioning system for finding things and people indoors," *IEEE Spectrum*, vol. 35, no. 9, pp. 71-78, Sept. 1998.

Vishal Bhargava received a BS degree in Electrical Engineering from the Indian Institute of Technology, Bombay and MS in Computer Networks from North Carolina State University. His research interests are in wireless networking and equity trading. He is currently working in Internet marketing.

Mihail L. Sichitiu was born in Bucharest, Romania. He received a B.E. and an M.S. in Electrical Engineering from the Polytechnic University of Bucharest in 1995 and 1996 respectively. In May 2001, he received a Ph.D. degree in Electrical Engineering from the University of Notre Dame. He is currently employed as an assistant professor in the Department of Electrical and Computer Engineering at North Carolina State University. His primary research interest is in wireless networking with emphasis on ad hoc networking and wireless local area networks.