Wireless LAN Location-Sensing for Security Applications

Ping Tao[†] ptao@ece.rice.edu
Algis Rudys[‡] arudys@cs.rice.edu
Andrew M. Ladd[‡] aladd@cs.rice.edu
Dan S. Wallach[‡] dwallach@cs.rice.edu

[†]Department of Electrical and Computer Engineering
[‡]Department of Computer Science
Rice University
Houston, TX

ABSTRACT

This paper considers the problem of using wireless LAN location-sensing for security applications. Recently, Bayesian methods have been successfully used to determine location from wireless LAN signals, but such methods have the drawback that a model must first be built from training data. The introduction of model error can drastically reduce the robustness of the location estimates and such errors can be actively induced by malicious users intent on hiding their location. This paper provides a technique for increasing robustness in the face of model error and experimentally validates this technique by testing against unmodeled hardware, modulation of power levels, and the placement of devices outside the trained workspace. Our results have interesting ramifications for location privacy in wireless networks.

CATEGORIES AND SUBJECT DESCRIPTORS

C.2.0 [**Computer Systems Organization**]: Computer-Communications Networks—*Security and protection*; C.2.1 [**Computer Systems Organization**]: Network Architecture and Design—*Wireless communication*; G.3 [**Mathematics of Computing**]: Probability and Statistics—*Markov processes,Probabilistic algorithms*; I.2.9 [**Computing Methodologies**]: Robotics—*Sensors*; I.5.1 [**Pattern Recognition**]: Models—*Statistical*

GENERAL TERMS

Algorithms, Design, Experimentation, Security

KEYWORDS

802.11, wireless networks, mobile systems, localization, probabilistic analysis

1 INTRODUCTION

IEEE 802.11b wireless LAN (WLAN) has been enthusiastically adopted in business offices, homes, hotels, cafés, and other spaces, both public and private, for wireless local network connectivity. WLAN has become a standard feature on laptop computers and is starting to appear in special-purpose consumer electronic devices. With this widespread deployment comes a danger when the network is abused. If an intruder connects to a traditional, wired network and begins transmitting packets, those packets can usually be physically traced to the port where they entered, and the rogue machine can be physically located and disconnected. In the wireless case, however, we only know that the rogue machine is associating with a given base station. This rarely provides enough information to physically locate the rogue machine. Moreover, wireless network technologies such as 802.11b are subject to additional classes of attacks that may be exploited by such a rogue machine [3], regardless of whether the network is using common forms of wireless encryption [25, 7, 17]. Because of this, determining the location of the rogue machine is a priority for administrators.

This problem of finding a rogue machine on a wireless network is a special case of the general wireless localization problem. The field of location-aware mobile computing and the field of mobile robot localization are both generally concerned with techniques to help various agents (whether human or machine) locate their position on a map. A wide variety of techniques have been designed to accomplish this, both using custom hardware devices, such as sonar or infrared sensors, and using the limited signal-strength measurements that can be performed by existing WLAN cards. Such a location sensing system has the potential to trump the inherent stealth advantage that intruders on wireless networks currently enjoy.

However, a rogue machine may desire *not* to be found. As a result, the localization must be performed by other agents in the system, such as customized base stations. The localizing agents do not know what WLAN hardware the rogue machine is using, and they do not know the power level at which the rogue machine is broadcasting. Indeed, the rogue machine could vary its broadcast power for every transmitted packet. In order to track such a target, a location sensing system must be sufficiently independent of the differences among mobile device configurations and must be able to overcome active interference by the intruder.

This paper presents a server-side indoor location-sensing system in which unmodeled variations in hardware and transmission power can be handled without significant degradation in localization precision. Our technique does not require any modifications to the hardware or software on the client being tracked. Our location-sensing system has been tested with a client which varies its transmission power, a possible evasive tactic for a rogue machine. Our experiments also demonstrate robustness against model error due to time-dependent variation in the quality of the wireless channel.

We begin with a discussion of related work in the field of wireless location-sensing. In Section 2, we discuss the methodology we

WiSE'03, September 19, 2003, San Diego, California, USA.

employed for determining a device's location. We describe the results from tracking experiments in Section 3. Section 4 contains a discussion of our results and future work. In Section 5, we present our conclusions.

1.1 Background

The field of location-aware computing [15, 10] deals with two principal tasks: determining and tracking the position of a mobile device, and providing useful user functionality based on a localization primitive. WLAN location sensing can determine the position of any laptop, PDA, or other device with WLAN hardware.

Our system and others like it use the signal strength readings from WLAN cards as a sensor and implement the Markov localization algorithm commonly used in various robotics applications [5, 11, 13, 26]. Following this technique, conditional probability distributions are built correlating sensor readings to position space by sampling these sensor readings at known positions in the building. This off-line phase is referred to as *training* or *learning*.

During the on-line phase, measurements are integrated and a probability distribution is built over position space. A maximum likelihood estimate is then used to determine position. A sequence of estimates can be integrated over time using various sensor fusion techniques. A Hidden Markov Model (HMM) can be used for this purpose [20, 18]. A general survey on probabilistic methods can be found in a comprehensive paper by Thrun [26]. Some interesting developments are discussed in a recent paper which compares several techniques experimentally [14]. A brief overview of these techniques as they relate to WLAN location sensing can be found in our original work [20].

Several early location-aware computing schemes used specialized hardware such as ultrasound transmitters or cameras to detect location [27, 23, 19]. Early schemes for wireless location sensing also relied on specialized transmitters or base stations [28, 16]. A number of systems have been built using probabilistic techniques to determine location based on RF signal strength for cellular telephone systems [21, 29]. The first system to use signal strength from off-the-shelf WLAN cards to detect location was RADAR [2, 1].

Recently, there has been a flurry of activity in applying probabilistic localization techniques to WLAN-based location sensing [9, 20, 30, 24, 31]. At the core of these various methods is the construction of conditional probability distributions relating sensor values to positions. This relation is constructed during a training phase as previously described. All these works reach similar conclusions, that robust one- to two-meter accuracy is achievable, that probabilistic methods effectively combat noise, and that it is difficult to automate training and parameter tuning. Operator, hardware, and transmission power variation is not discussed in any of these works.

2 METHODOLOGY

In this section, we discuss our methodology for determining a user's location using signal strength readings from WLAN cards. We begin by describing our experimental setup. We then discuss our observations of signal propagation properties in an indoor environment. Finally, we discuss our algorithms for determining a user's location.

2.1 System setup

In our prior work on WLAN localization [20], we had a mobile laptop measuring the signal strengths from fixed access points (APs). An initial training phase took measurements at positions spaced approximately every 1.5 meters on the third floor of Duncan Hall, the building housing Rice University's Computer Science department. This training data was used to create a Bayesian network that could take subsequent observations of signal strengths and yield a probability distribution of where the mobile device might be in the building.

Our previous implementation measured the signal strengths of APs as observed by the laptop. This decision is arbitrary; the location-sensing algorithm we presented can be implemented with the client as the observer or with the APs as observers. The implementation presented in this paper measures signal strength from the APs. This server-side architecture has the advantage of allowing us to localize a laptop independent of any specialized hardware or software on the laptop.

Our system consists of a centralized server and a number of snoopers. Snoopers provide overlapping coverage for the target area, much like APs do in a production network. During training and localization, the server notifies the snoopers of the target media access control (MAC) address, the channel number, and the listening period. The snoopers will then record the signal strength of all packets received that match the server's query and transmit those results to the server, which can then infer the physical location of the target using algorithms described in Section 2.3.

In our current prototype, the server communicates with the snoopers using our preexisting in-building WLAN. In a future production environment, the snooper functionality might be integrated directly into WLAN APs.

2.1.1 Hardware

Snoopers. Snoopers are responsible for observing the signal strength of packets transmitted by the target machine. In our experiments, we used five laptops from various manufacturers, all running Windows XP and using D-Link AirPlus DWL-650+ WLAN PCMCIA cards containing the Texas Instruments ACX100 single-chip 802.11b WLAN implementation. We wrote a customized device driver, allowing us to extract the signal strength, signal-to-noise ratio, and signal-to-interference ratio of each received packet. So far, we have only used the signal strength, as in our previous work, although the other information might be useful for increasing localization accuracy in future work.

The signal strength reading is taken from the receiver's automatic gain control (AGC) register. The controller updates this 8-bit register for every individual packet with the signal strength of the packet's physical layer header. This packet header is sent at a constant data rate independent of the transmission data rate of the packet body. We do not have any concrete understanding of how the AGC register values map to the actual signal intensities.

Where a normal WLAN AP will only receive packets from "associated" stations, our customized driver allows our snooper to listen to all traffic on any given channel. Upon request, it temporarily switches channel, measures the target station's signal strength, and switches back to resume normal network operations. We use this technique to allow the snoopers to perform tracking and communicate with the central server through the same WLAN card. A similar technique was developed for RADAR [1].

Target machine. For training and testing, we used a Dell Latitude X200 sub-notebook. The X200 has a built-in antenna to support an internal Mini PCI WLAN card. A normal PCMCIA slot is also available. Most of our tests and training used a Mini PCI version of the aforementioned D-Link card with the internal antenna. We used a custom device driver with this card, allowing us to vary the card's transmission power. To ensure that our results

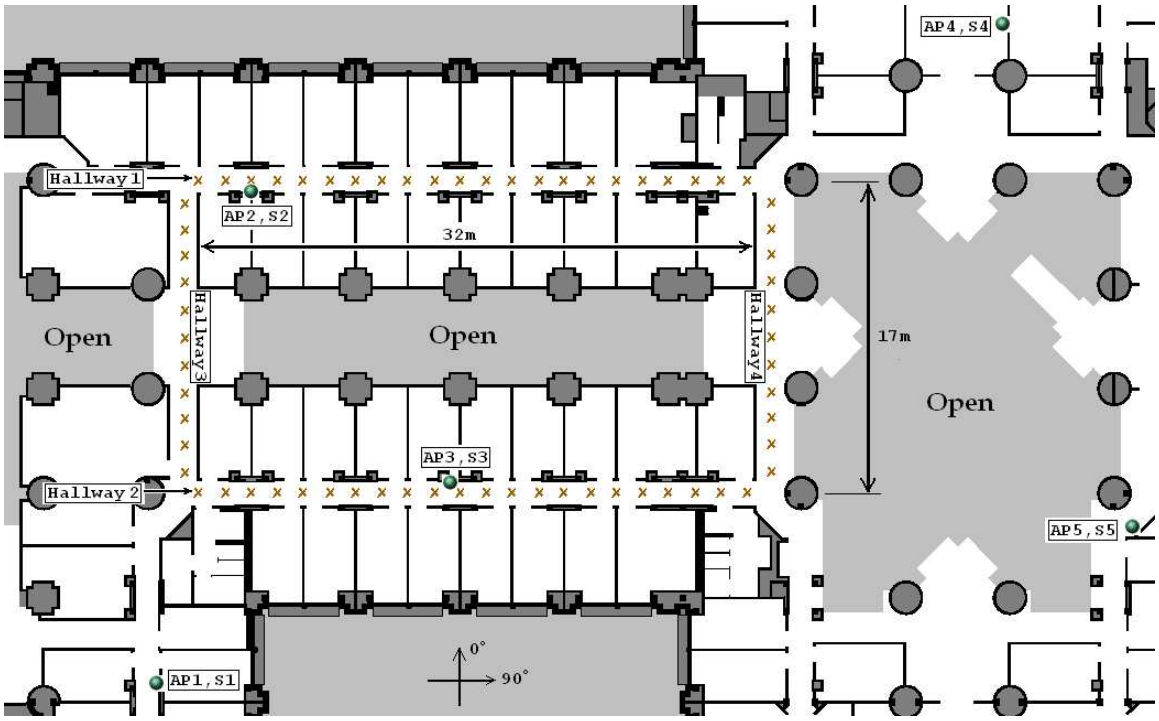

Figure 1: Map of the region of the Duncan Hall where we conducted our tests. There are 5 wall-mounted access points (AP1-5) providing overlapping coverage for this area. We placed one snooper (S1-5) under each AP to measure the signal strength from the target machine. The *x* marks on the map are the points where the training took place.

were independent of this particular card, we also used a Linksys WPC11 PCMCIA card, which has an antenna built into the card. The Linksys card uses the Intersil Prism2 chipset, a completely different WLAN implementation. Accordingly, it also uses a different set of software drivers.

Server. The server is a Java program that communicates with the snoopers to collect signal strength measurements on packets observed from the target machine. The server needs sufficient memory and processing power to contain the Bayesian network. In practice, any modern laptop has more than enough power to track a single user in real time. In a production environment, the server might be tracking large numbers of users simultaneously and would need to run on a dedicated machine.

2.2 In-building RF signal propagation

We conducted our experiments on the third floor of Duncan Hall at Rice University, using four hallways as shown in Figure 1. Hallways 1 and 2 are narrow long enclosed hallways with fiberglass ceiling tiles, carpeted concrete floors, and painted sheet rock walls with occasional concrete structural pillars. Hallways 3 and 4 are open to the ceiling of the building some 30 feet overhead. Furthermore, hallway 4 is adjacent to an open-air atrium overlooking the building's lobby. While the accuracy of our techniques would certainly differ in other buildings, we believe our building offers a diversity of materials and architectural styles that provides a significant challenge to accurate localization.

2.2.1 Channel variations

Our previous work directly used the signal strength histogram obtained at each training point to infer a user's location, based on the observation that the signal strength distributions were non-Gaussian, and thus did not necessarily yield meaningful "average" values. By doing this, we essentially assumed that the histograms were not changing over time. However, subsequent experiments, shown in Figure 2, show that the signal strength histograms vary noticeably as a function of the time of day, with significantly more noise while more people are in the building. We also observe that the histogram from the first night had two modes, while the histogram from the second night had only one mode. From these and other similar measurements, we have concluded that the average signal strength is the only robust value to use across different days and times of the day.

2.2.2 Transmission power

A key problem for localizing rogue machines is being robust in the face of different WLAN implementations and variation in the transmission power of individual packets. Figure 3 shows how the observed signal strength changes as transmission power is varied. We observe that the relative ordering of observed signal strengths remains constant. Moreover, our experiments suggest that observed signal strength is linearly proportional to transmission power. Most importantly, the differences in received signal strengths do not vary dramatically as the transmission power changes. We will use these observations to design a filter to improve our localization robustness, as described in Section 2.3.2.

2.3 Algorithms

We tested two different localization algorithms to compare their suitability in localizing a rogue mobile node. The first, the *Histogram* method, is the Bayesian inference algorithm developed in our prior work [20]. The second, the *Difference* method, is our new

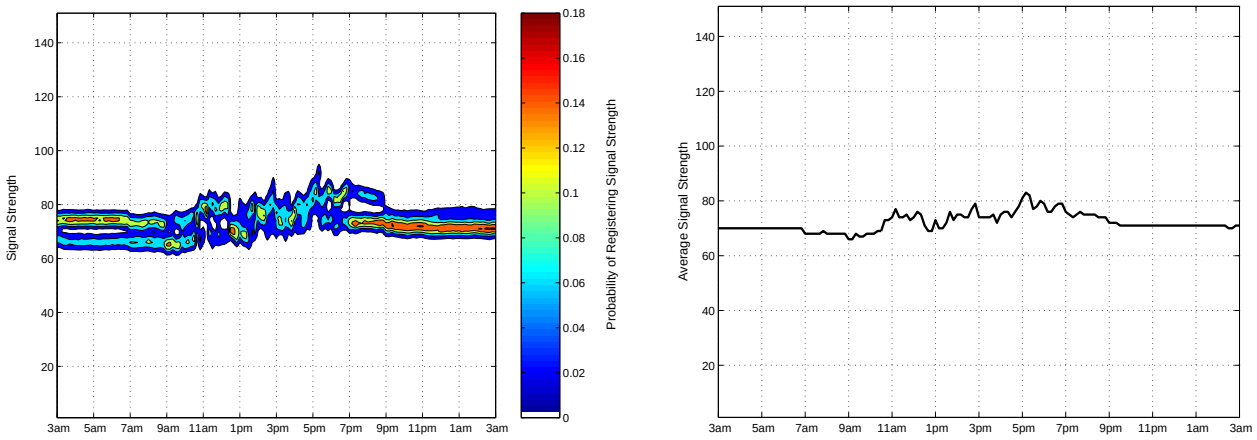

Figure 2: Observed signal strength histogram variation over time from a laptop to a fixed AP.

robust localization algorithm. The Difference method is a weighting heuristic loosely based on Bayesian inference.

2.3.1 Histogram method

This section summarizes our original localization method, which we compare against our new technique. Although the algorithm we used is unchanged, it was deployed in a slightly different way. As discussed in Section 2.1, the signal strength of the laptop is now measured at fixed base stations, instead of vice-versa. In addition, there are fewer fixed base stations than we had at our disposal in our original work.

The histogram method uses a Bayesian inference scheme to locate a WLAN user. We model the world as a finite *position space* $\{p_1,\ldots,p_n\}$ with a finite *observation space* $\{o_1,\ldots,o_m\}$. The *sensor model* $Pr(o_j|p_i)$ is a learned model of the conditional probability of seeing observation o_j at position p_i. A *position vector* π is a probability vector over the various positions (i.e., π_i represents the probability that position p_i is the current position). Given a prior estimate π, after observing o_j we can estimate our new position vector π' by calculating the individual conditional probabilities π'_i for each $i \in \{1,\ldots,n\}$ using Bayes' rule,

$$\pi'_i = \frac{\pi_i \cdot Pr(o_j|p_i)}{\sum_{\alpha=1}^{n} \pi_\alpha \cdot Pr(o_j|p_\alpha)} .$$

We combine these π'_is into the new estimate of our position, π'. We then choose the most likely position as the representative position from this position vector. Each position p_i is a tuple $(x_i, y_i, z_i, \theta_i)$ describing a user's location and orientation. During the training phase, for each p_i, we collect signal strength measurements from the snoopers. This raw data can be reused to train multiple localization systems, each of which will then define its own mapping from observations to positions, i.e., the probability distribution $Pr(o_j|p_i)$.

We define an observation as a vector of signal strength readings over k snoopers,

$$o_j = (\lambda_1,\ldots,\lambda_k) ,$$

where λ_ρ is the signal strength measured by snooper ρ. We then define $Pr(o_j|p_i)$ as

$$Pr(o_j|p_i) = \prod_{\rho=1}^{k} Pr(\lambda_\rho|p_i) .$$

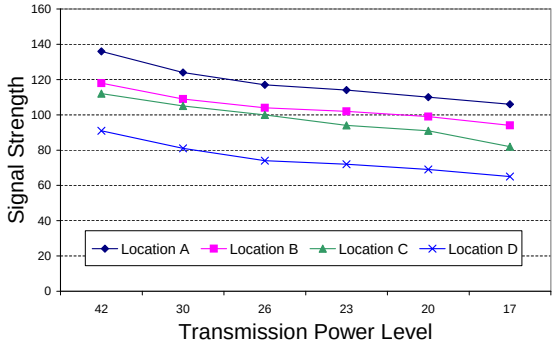

Figure 3: Signal strength readings from four different receivers of signal from a single transmitter, with the transmitter varying its transmission power. Higher values on the x-axis reflect higher transmission power. Higher values on the y-axis reflect a stronger signal at a receiver.

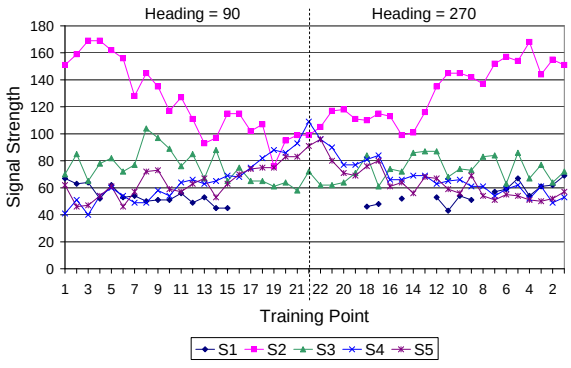

Figure 4: Signal Strength Variations in Hallway 1 at 22 positions for each of two orientations. Missing segments for S1 indicate that Snooper 1 did not receive any packet from the target machine at those positions.

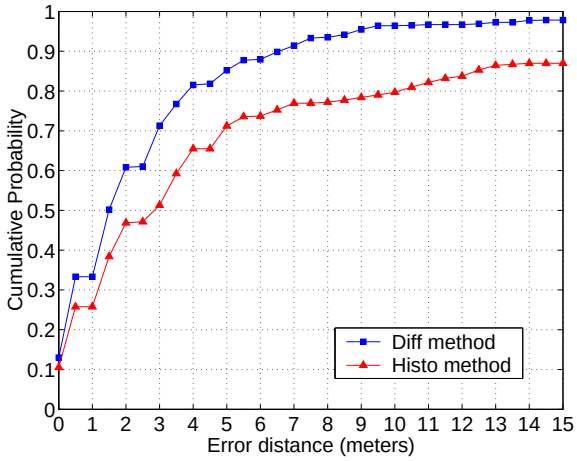

Figure 5: Cumulative error over the four hallways, using the same WLAN card for training and for localization.

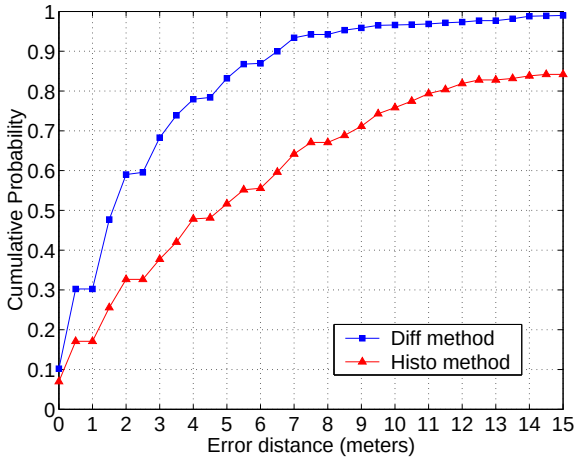

Figure 6: Cumulative error over the four hallways, where the transmission power level is reduced when localization occurs.

This method directly uses the signal strength histogram obtained from training to get each $Pr(\lambda_\rho|p_i)$, so we call it the *Histogram* method or briefly *Histo*.

As we have shown in Section 2.2.1, the signal strength histogram changes over time, and so might not be reliable for localization. As our results in Section 3 show, the *Histo* method is sensitive to changes in the model, and performs poorly in the face of model error.

2.3.2 Difference method

Our *Histo* method assumes that the trained signal strength histogram accurately models the signal strengths that will be later observed. This is not necessarily true when the target client is using a WLAN card different from the one with which we trained the system, or when the target client is intentionally altering its transmission power. In order to accommodate such variations, we developed another localization algorithm based on our observations in Section 2.2.2. Since lower transmission power tends to cause linear decreases in all observed signal strengths, we can use the *differences* between the observed signals rather than the raw signals themselves.

We post-process our training data to be the differences in signal strength between every pair of snoopers, fitted to the Gaussian distribution $N(\mu,\sigma^2)$ with the average μ being the average difference in signal strengths. The standard deviation σ was chosen as 12 to accommodate sampling errors and the variations of average signal strength over the course of a day, as shown in Figure 2.

During localization, as each snooper receives a packet and reports the signal strength to the localization server, the server computes the difference in signal strength of reports between every pair of snoopers. During each inference window, the server receives several packets and can then compute the difference in average signal strength $(\overline{\lambda_{\rho_\alpha}} - \overline{\lambda_{\rho_\beta}})$ for every pair of snoopers ρ_α and ρ_β. The statistics generated from all packets received during this inference window define an observation.

We found that using a weighting scheme where the conditional probability of each difference in signal strength was added to the probability for that location gave the best accuracy. We again model the world as a finite position space $\{p_1,\ldots,p_n\}$. The weights $W(p_i)$ were computed as follows

$$W(p_i) = \sum_{\alpha=1}^{k-1} \sum_{\beta=\alpha+1}^{k} Pr(\overline{\lambda_{\rho_\alpha}} - \overline{\lambda_{\rho_\beta}}|p_i) \; .$$

Once the weights have been calculated for each p_i, we choose the position with the largest weight as our location estimate. We call this algorithm the *Difference* method or briefly *Diff*.

3 Results

In this section, we examine the accuracy of our localization system in a number of different scenarios. We first evaluate straightforward localization using the Mini PCI D-Link card with which we trained the system. We then consider localization of a hypothetical rogue machine that varies its transmission power to avoid detection. We also examine localization when using a different WLAN card from the one which we used to train the system. Finally, we discuss localization of points outside the training region.

3.1 Experimental setup

We conducted our experiments on the third floor of Duncan Hall at Rice University, using four hallways as shown on the map in Figure 1. We placed 5 snoopers (S1 through S5) under the five APs shown in map. While we could have placed them anywhere, this allowed us to simulate what our current APs could do if they were augmented to support snooping. It also represented a perfectly reasonable distribution to observe all packets on the third floor. Our snoopers are between 22 and 50 meters away from each other, with no line of sight between any two snoopers.

During the training phase, we measured signal strengths at positions every 1.42 meters (56 inches) from one end of each hallway to the other end facing in both directions. At each training point, we took a trace at 5 samples per second until the signal strength histogram converged, sometimes taking as long as one minute per point. This trace data was all taken using the default transmission power level, and was used to train both our *Histo* and *Diff* systems, as described in Section 2.3.

As an example of the training data we obtained, Figure 4 shows the average signal strength recorded for each training point in hallway 1. Although we actually captured the full histogram of signal strengths observed from each location, we present the average of

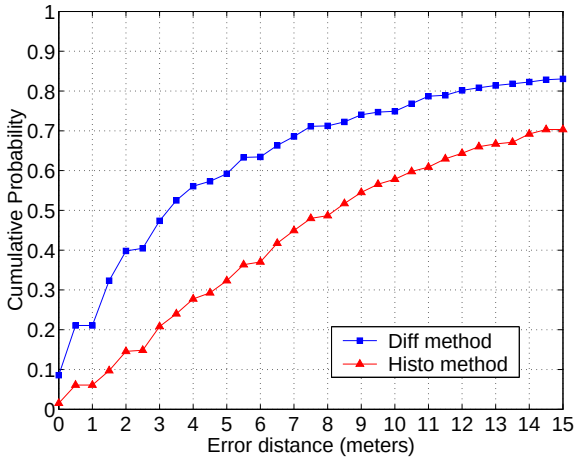

Figure 7: Cumulative error over the four hallways, where the transmission power level changes for every packet transmitted.

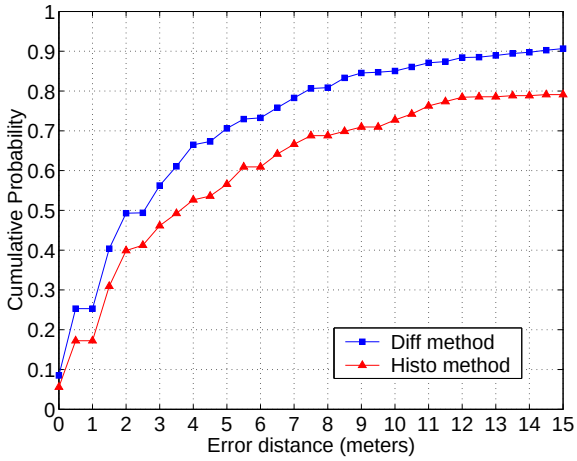

Figure 8: Cumulative error over the four hallways, where different WLAN cards were used for localization and training.

those values to simplify the graph. This graph shows a clear trend in signal strength variation as we moved from one end of the hallway to the other, and it also shows that changing orientation yields different signal measurements.

Each experiment consisted of walking around the loop formed by the four hallways of our test area in a counterclockwise direction, down hallways 1, 3, 2, and 4 in that order. At each training point and between every two training points, we took a 15-second trace of data, which was input into the *Histo* and *Diff* localization inference engines to generate a series of position estimations.

3.2 Basic localization

When we localized using the same WLAN card as we had used in training, our accuracy was quite high. Over the four hallways, the localization error was at most 2 meters 61% of the time for the *Diff* method. The *Histo* method only achieved 2 meter accuracy 47% of the time. These results are illustrated in Figure 5.

Figures 9 through 12 show cumulative error for the four hallways individually. For all hallways except the third, the *Histo* method and the *Diff* method are nearly indistinguishable in terms of accuracy. The *Histo* method has an error of at most 2 meters for 51%, 70%, and 56% of sampled positions, respectively, for hallways 1, 2, and 4. The *Diff* method has an error of at most two 2 meters for 54%, 66%, and 51% of positions, respectively, for the same hallways.

For hallway 3, *Diff* turns in one of its best performances, with an error of at most 2 meters at 64% of sampled positions. *Histo*, by contrast, turns in its worst accuracy, with fewer than 15% of location estimates being within 2 meters and fewer than 39% of estimates even within 15 meters (this on a hallway 17 meters long). In our experimental configuration, *Histo* frequently reports hallway 3 as hallway 1, while *Diff* correctly distinguishes the two hallways. Indeed, while *Diff* consistently performs very well on hallway 3, *Histo* consistently performs extremely poorly.

3.3 Varying transmission power level

The robustness of *Diff* relative to *Histo* becomes evident when localizing a transmitter which transmits at a reduced power level. Our results show the *Diff* method achieving nearly the same accuracy (2 meters, 59% of the time) with lower-power transmission as it achieved with default-power transmission. The *Histo* method achieved 2 meter accuracy only 33% of the time. The results are illustrated in Figure 6.

We ran another test in which we varied the transmission power randomly for each transmitted packet. We achieved accuracy of 2 meters 40% of the time using our *Diff* method. The *Histo* method achieved 2 meters accuracy only 15% of the time. The results show that while an attacker may have some success varying her transmission power, the *Diff* method vastly improves our ability to detect her. The results are illustrated in Figure 7. Causing the transmission power of the WLAN card to vary wildly is also stressful on the card itself, and we lost one card to burn-out because of this.

3.4 Varying transmitter

We next experimented with localizing our laptop using a Linksys WLAN card, having a different antenna and different chipset from the D-Link card used in training. We achieved comparable accuracy between *Diff* and *Histo*. With *Diff*, an error of at most 2 meters was achieved at 49% of the sampled positions. *Histo* achieved this accuracy for 40% of the sampled positions. These results are illustrated in Figure 8.

3.5 Untrained poses

Any localization method which relies on training can only localize targets to points within the trained area. Neither the *Diff* nor the *Histo* method will correctly localize a mobile device anywhere outside of the four hallways. Furthermore, small deviations from the trained positions can confuse a training-based localization system. We tested how the two algorithms compared when localizing a laptop in untrained positions.

We first ran two traces of untrained positions along the training hallways. The first trace involved localizing a laptop which was very close to the wall in a trained hallway; the original training data was taken in the center. *Diff* achieves an accuracy of 2 meters at 49% of sampled points (versus 60% in the center). This is similar accuracy to other cases which break the model, including the low-power and varying transmitter tests. By comparison, *Histo* performed as well as in the simple localization case, achieving an accuracy of 2 meters at 46% of sampled points. The results are illustrated in Figure 13.

We also tried to localize a laptop which was facing perpendicular to a trained direction. The results are comparable to the earlier

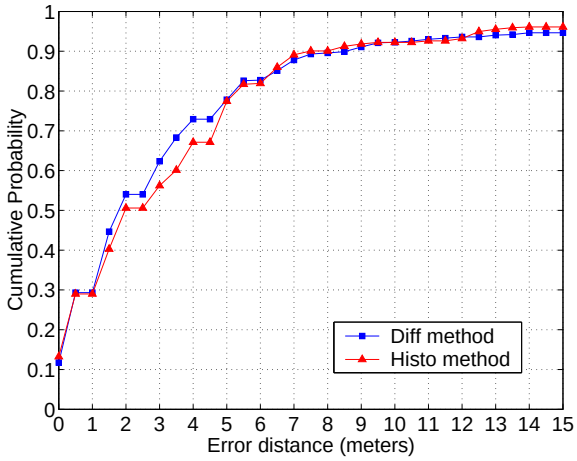

Figure 9: Cumulative error for hallway 1, using the same WLAN card for training and for localization.

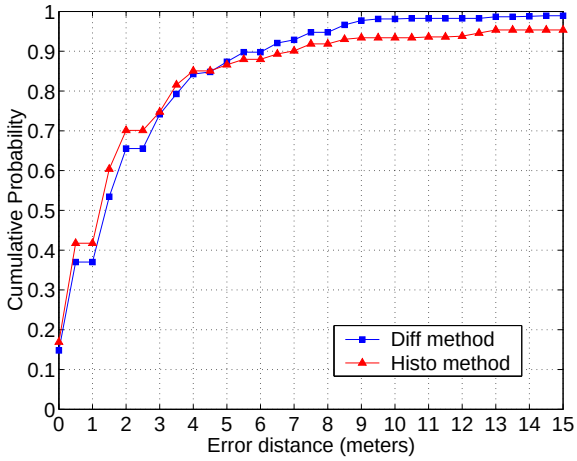

Figure 10: Cumulative error for hallway 2, using the same WLAN card for training and for localization.

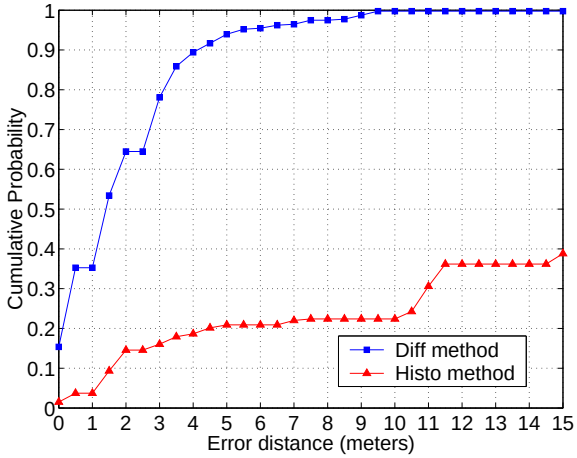

Figure 11: Cumulative error for hallway 3, using the same WLAN card for training and for localization.

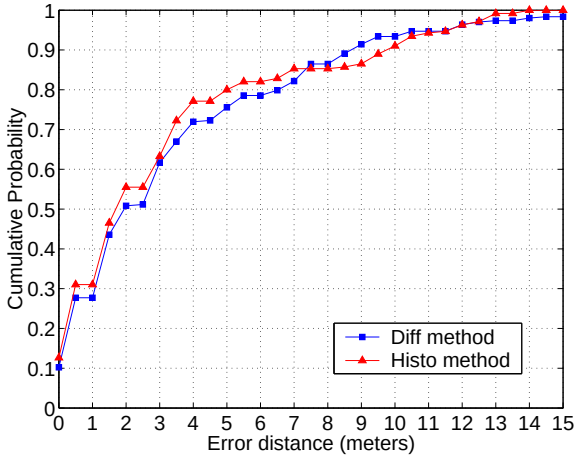

Figure 12: Cumulative error for hallway 4, using the same WLAN card for training and for localization.

results for localizing close to a wall. *Diff* localized to within 2 meters 50% of the time, and *Histo* localized to within 2 meters 46% of the time. These close results show that *Histo* is more robust against certain forms of model variation, but that *Diff* outperforms *Histo*, regardless. The results are shown in Figure 14.

Finally, we wanted an idea of how these algorithms performed when the target is not even in one of the trained hallways. To determine this, we took measurements at a number of sample points outside of our training area. We sampled at seven test positions on the third floor, four positions in rooms along the trained hallways and three positions in nearby hallways. In all of those seven tests, the *Diff* method estimated the location a nearby position within the training area. The *Histo* method estimated most of the positions correctly, but at two positions, one in an adjacent room and one in a hallway, it guessed the wrong side of the building.

We also tested several positions on the second floor of Duncan Hall. For position below hallways 3 and 4, the *Diff* method localized to the correct respective hallways, although not always directly above. The *Histo* method correctly identified hallway 4, but missed hallway 3. For positions below hallway 1, by comparison, most of the *Diff* method's estimates were off by a large distance, while the *Histo* method localized to hallway 1 for most of the samples.

Although there is no hallway on the second floor directly below hallway 2, there is a hallway parallel to hallway 2 on the south side of the building that is open on one side to the open area that cuts across the middle of the building. Attempting to localize positions on that hallway, the *Diff* method concluded that we were on hallway 3. Although we were physically closer to hallway 2, we had line-of-sight with hallway 3, so this conclusion is not surprising. The *Histo* method, by contrast, consistently concluded that we were on hallway 1. This is understandable, considering *Histo*'s tendency to confuse hallways 1 and 3, as we discussed in Section 3.2.

Of course, no training-based method will return locations outside of its training set. However, as these results show, we can often derive points in the original training set that are meaningfully "close" to the target's actual location. Therefore, in practice, it should be sufficient to train exclusively in major hallways and open areas to yield useful localization results, even if the target machine is actually in a side office.

3.6 Variations on Diff

There are a number of ways to calculate weights based on relative signal strength. We evaluated a total of three. In the first method, which is described in Section 2.3.2, the difference in signal strength

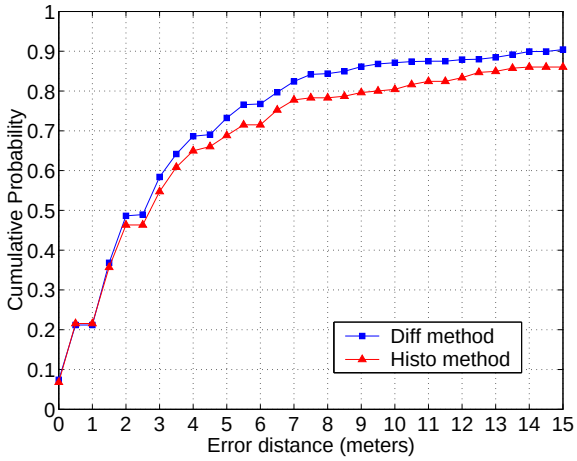

Figure 13: Cumulative error over the four hallways when the target is very close to a wall along the training hallways.

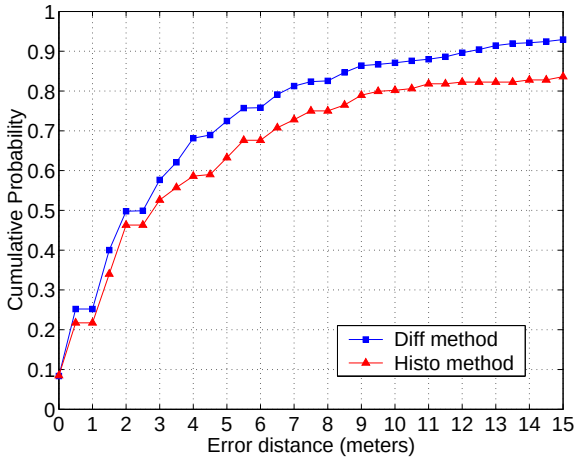

Figure 14: Cumulative error over the four hallways when the target is facing at 90° to the direction of the hallway (an untrained direction).

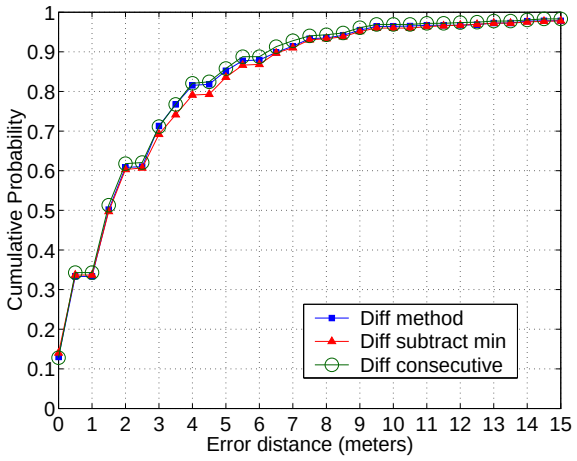

Figure 15: Cumulative error for the three *Diff* methods over the four hallways, where the target machine and WLAN card are the same as used in training.

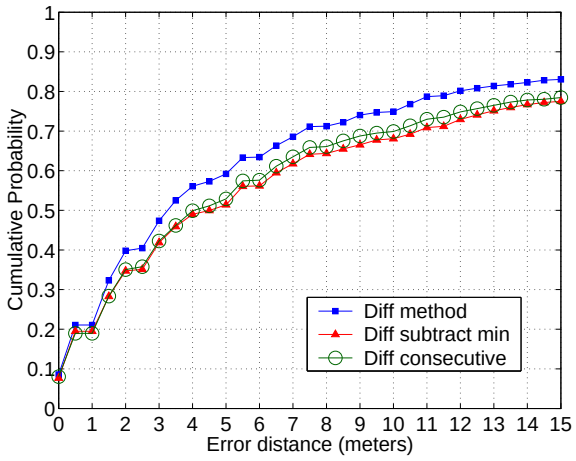

Figure 16: Cumulative error for the three *Diff* methods over the four hallways, where the transmission power level changes for every packet transmitted.

between every pair of base stations is used in the localization algorithm. A second scheme we tried was to use the difference between each base station signal strength reading and the minimum signal strength reading in the localization algorithm. The final scheme we evaluated was computing the difference in signal strength between each base station and the base station after it taken in the order that the base stations appear in the trace.

Figure 15 shows the cumulative error for localization performed with the same card as trained. All the *Diff* variants are very close. The test of using a different transmitter from training to localization shows similar results. Figure 16 shows the cumulative error for localization performed with the transmitter varying its power level. In this test, while all of the *Diff* variants have very similar accuracy, the original *Diff* method edges out the others. The low-power localization test shows similar results.

4 DISCUSSION

This section discusses a number of issues and implications of wireless location systems, both as we have designed and in general.

4.1 Accuracy versus robustness

We have presented two methods for wireless localization, *Histo* and *Diff*. The former uses a rigorous and fairly well-understood model for inferring location from sensor data. When *Histo* works, it works well. *Histo* is also fairly robust when the pose of the target node varies from what was trained.

However, *Histo* is very sensitive to other aspects of the training model. In particular, since *Histo* depends on the accuracy of the signal strength histograms, when an attacker varies the broadcast power on each packet, *Histo* gets lost. *Diff*, on the other hand, can gracefully handle such variation, and performs reasonably well in other cases.

Of course, the accuracy of both algorithms could be improved by better distribution of the snooper nodes, by placing more snooper nodes in the building, and by tuning parameters inside the algorithms. Regardless, when the localization system can make stronger model assumptions, such as assuming that all wireless nodes are using the same WLAN implementation and broadcasting at full power, it will generally be able to achieve higher precision. However, if those assumptions prove to be false, the robustness of

the localization will necessarily suffer relative to what could have been achieved with a model making fewer such assumptions.

In cases where it really matters, there is no reason the system could not perform both localization algorithms. Nodes that *want* to know their location, and thus presumably follow the rules, can get an accurate answer. Administrators trying to track down the physical location of a malicious node, on the other hand, can use the more robust tracker, which will certainly help narrow the physical search space.

4.2 Comparison to prior work

In our prior work [20], we consistently saw better accuracy than observed here. Several factors worked against us this time. First, in the tests described here, only five snoopers were available, and generally no more than four had visibility of the target at any given position. In the previous work, up to nine APs were used for localization at any position. The additional APs could be used to disambiguate certain points in the training area, helping to increase measurement accuracy.

Furthermore, in this work, we performed no sensor fusion. In our prior work, the output of our Bayesian inference engine was post-processed by a filter that fused multiple location samples over time based on some simple assumptions. These assumptions included observing, based on a simple probabilistic model of human movement [5], that the object being tracked is unlikely to make large, discontinuous jumps across the building. We found that such techniques significantly improved localization accuracy, and they would certainly apply here.

4.3 Privacy concerns

We have presented a methodology for localizing nodes within a building *without their cooperation*. If a node transmits a packet inside a building that has wireless snoopers and a trained localization engine, that node can be localized. While the original motivation for this work is allowing administrators to locate nodes that actively misbehave and remove them, it can also allow for less agreeable uses. For example, an employer could track the locations of its employees, or a shopping mall could track the locations of its shoppers.

Furthermore, the owner of the building need not be the operator of the snoopers. Rogues could install their own snoopers (e.g., placing laptops above the ceiling tiles) and localize nodes within the building without the knowledge or consent of the building's owners or occupants. This creates a clearly undesirable situation, defeating any attempts at restricting the disclosure of presumably sensitive location information (e.g., Canny [8]). Our work implies that anyone who can deploy snoopers and train their localization system will be able to localize nodes in the area, even if those nodes take steps to obfuscate their transmission power.

The only exception to this may be nodes that are physically distant from the wireless network to which they are connected. An attacker who is *war driving* outside the building may well be able to interact with the in-building wireless network; such attackers might also use parabolic antennas, further increasing their physical separation [6, 12]. Attackers with such unusual high-power antennas may well generate localization data sufficiently unlike our normal training data as to be virtually unlocalizable.

4.4 Future work

There is still progress to be made in improving wireless localization. By computing the differences between sampled signal strengths, we are able to filter out variations that can occur as a result of varying transmission power, but this is just one instance of a more general space of possible preprocessing filters. Future studies might spend more effort on applying a preprocessing filter to the sample data before feeding it to an inference engine. Likewise, our work does not consider any postprocess filtering or sensor fusion of localization inferences. Finally, difference in signal strength could be integrated into a Bayesian inference algorithm for wireless localization. These and many other techniques may be able to significantly improve the accuracy and robustness of localization.

Our work suggests that it may be very difficult to physically hide a wireless network node that is actively broadcasting packets. However, we did not consider *coalitions* of hostile nodes, working in concert to attack a network. Such coalitions might be able to confuse the localization system by presenting the illusion of an attacker being simultaneously in multiple locations, particularly if some of the hostile nodes are actively moving. A more robust localizer might try to solve clustering problems to determine the number and locations of hostile nodes.

Likewise, while nodes which wish to communicate normally on a network generally need to maintain a constant MAC address, an attacker has no such constraints. In order for our localization system to operate effectively, it must be able to distinguish "evil" packets from normal packets [4]. This is generally the domain of Network Intrusion Detection Systems such as Bro [22]. Building a suitable packet classifier for wireless network misbehavior would require knowing something about how an attacker may or may not choose to misbehave. In practice, it might be preferable for the classifier to err on the side of false positives once an attack is under way, assuming suitable clustering algorithms can post-process the localization data. If an attacker's probable location can be reduced to a small enough number of possibilities, that may be acceptable to the administrator seeking to stop the attacker.

A further concern is attackers operating at a significant distance, using parabolic or otherwise non-standard antennas. An interesting question is whether the in-building snooper network could be trained to identify, at a minimum, the compass direction from the building to the attacker. Such information could greatly reduce the effort necessary to find the attacker.

5 Conclusion

Traditional localization methods tend to have simple models of how nodes will behave. Malicious nodes can easily violate these assumptions by modulating their transmission power for each packet. We present a mechanism for locating mobile devices in an indoor environment, even when the nodes might be malicious. Our techniques sacrifice some amount of accuracy in the ideal case of localizing cooperative nodes, but maintain robustness when faced with a variety of model errors, including malicious nodes, nodes with different hardware than we trained against, and, to some extent, nodes located outside of the training area. We conclude that wireless nodes can be localized whether or not they wish to be, raising interesting privacy issues in the use of wireless networks.

6 Acknowledgements

The authors wish to thank Kostas Bekris, Guillaume Marceau and Lydia Kavraki for assistance with an earlier version of this work. Dave Johnson, Ed Knightly, and Shu Du also offered valuable advice and assistance.

This work is supported in part by NSF Grant CCR-9985332 and Texas ATP grant #03604-0053-2001 and by gifts from Microsoft and Schlumberger. Andrew Ladd is partially supported through his advisor, Lydia Kavraki, by NSF IRI-9702288, NSF ACI-0205671,

a Whitaker Grant, and a Sloan Fellowship. Andrew Ladd is also partially supported by an FCAR grant.

7 REFERENCES

[1] P. Bahl and V. N. Padmanabhan. Enhancements to the RADAR user location and tracking system. Technical Report MSR-TR-2000-12, Microsoft Research, Feb. 2000.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proc. of IEEE Infocom 2000*, volume 2, pages 775–784, Tel Aviv, Israel, Mar. 2000.

[3] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the USENIX Security Symposium*, Washington, DC, Aug. 2003.

[4] S. Bellovin. The security flag in the IPv4 header. RFC-3514, Internet Engineering Task Force, Apr. 2003. `http://www.ietf.org/rfc/rfc3514.txt`.

[5] W. Burgard, A. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proc. of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, Madison, WI, July 1998.

[6] S. Byers and D. Kormann. 802.11b access point mapping. *Communications of the ACM*, 46(5):41–46, May 2003.

[7] N. Cam-Winget, R. Housley, D. Wagner, and J. Walker. Security problems in 802.11-based networks. *Communications of the ACM*, 46(5):35–39, May 2003.

[8] J. Canny. Some techniques for privacy in Ubicomp and context-aware applications. In *Workshop on Socially-informed Design of Privacy-enhancing Solutions in Ubiquitous Computing*, Göteborg, Sweden, Sept. 2002.

[9] P. Castro, P. Chiu, T. Kremenek, and R. Muntz. A probabilistic location service for wireless network environments. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp)*, Atlanta, GA, Sept. 2001.

[10] G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical Report TR-2000-381, Department of Computer Science, Dartmouth College, Nov. 2000.

[11] G. Dudek and M. Jenkin. *Computational Principles of Mobile Robotics*. Cambridge University Press, Cambridge, UK, 2000.

[12] J. Duntemann. *Drive-By Wi-Fi Guide*. Paraglyph Press, Scottsdale, AZ, Feb. 2003.

[13] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research, (JAIR)*, 11:391–427, Nov. 1999.

[14] J.-S. Gutmann and D. Fox. An experimental comparison of localization methods continued. In *Proceedings of International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, Sept. 2002.

[15] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, Aug. 2001.

[16] J. Hightower, R. Want, and G. Borriello. SpotON: An indoor 3D location sensing technology based on RF signal strength. Technical Report UW CSE 00-02-02, Department of Computer Science and Engineering, University of Washington, Seattle, WA, Feb. 2000.

[17] R. Housley and W. Arbaugh. Security problems in 802.11-based networks. *Communications of the ACM*, 46(5):31–34, May 2003.

[18] L. Kaebling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[19] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for EasyLiving. In *Third IEEE International Workshop on Visual Surveillance*, Dublin, Ireland, July 2000.

[20] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach. Robotics-based location sensing using wireless Ethernet. In *Proceedings of The Eighth ACM International Conference on Mobile Computing and Networking (MOBICOM)*, Atlanta, GA, Sept. 2002.

[21] T. Liu, P. Bahl, and I. Chlamtac. Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks. *IEEE Journal on Selected Areas in Communications*, 16(6):922–936, Aug. 1998.

[22] V. Paxson. Bro: a system for detecting network intruders in real-time. *Computer Networks*, 31(23–24):2435–2463, 1999.

[23] N. Priyantha, A. Miu, H. Balakrishman, and S. Teller. The Cricket compass for context-aware mobile applications. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 2001)*, pages 1–14, Rome, Italy, July 2001.

[24] T. Roos, P. Myllymaki, H. Tirri, P. Misikangas, and J. Sievanan. A probabilistic approach to WLAN user location estimation. *International Journal of Wireless Information Networks*, 9(3), July 2002.

[25] A. Stubblefield, J. Ioannidis, and A. D. Rubin. Using the Fluhrer, Mantin, and Shamir attack to break WEP. In *2002 Network and Distributed Systems Symposium*, San Diego, CA, Feb. 2002.

[26] S. Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 21(4):93–109, 2000.

[27] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, Oct. 1997.

[28] J. Werb and C. Lanzl. Designing a positioning system for finding things and people indoors. *IEEE Spectrum*, 35(9):71–78, Sept. 1998.

[29] R. Yamamoto, H. Matsutani, H. Matsuki, T. Oono, and H. Ohtsuka. Position location technologies using signal strength in cellular systems. In *Proc. of the 53rd IEEE Vehicular Technology Conference*, Rhodes, Greece, May 2001.

[30] M. Youssef and A. Agrawala. Small-scale compensation for WLAN location determination systems. In *Proceedings of IEEE Networking and Communications Conference*, New Orleans, LA, Mar. 2002.

[31] M. Youssef, A. Agrawala, and A. U. Shankar. WLAN location determination via clustering and probability distributions. In *Proceedings of IEEE Conference on Pervasive Computing and Communications*, Fort Worth, TX, Mar. 2003.