A Software System for Locating Mobile Users: Design, Evaluation, and Lessons

Paramvir Bahl and Venkata N. Padmanabhan
Microsoft Research
{bahl, padmanab}@microsoft.com

Anand Balachandran
University of California at San Diego
anandb@cs.ucsd.edu

Abstract

We have built a software system, RADAR, to locate mobile users connected to an in-building radio-frequency (RF) wireless LAN. RADAR uses signal strength information extracted from the wireless network interface, in conjunction with a Radio Map of the building, to determine location. Over the past year we have deployed this system in multiple buildings on our campus using two different wireless LAN technologies and two widely used operating systems. This experience has led us to identify some shortcomings of the basic RADAR system and fundamental limitations in the way wireless network hardware is abstracted in contemporary operating systems.

To address the shortcomings, we present several novel algorithms, including an environmental profiling algorithm to compensate for the vagaries of the RF environment, and a history-based algorithm for continuous user tracking. Performance evaluation of these algorithms using extensive data collected from our deployments shows that the accuracy of pinpointing a mobile's location improves by up to a factor of 3 in some cases. In addition, we have developed WiLIB, a hardware-agnostic library that exposes to user-level applications facets of the wireless network (e.g., received signal strength) that are not exposed by current operating systems. We discuss our experience in implementing RADAR over WiLIB and also additional ways in which WiLIB can enable novel wireless applications.

RADAR is implemented purely in software and is easily deployable over a standard wireless LAN.

1 Introduction

The proliferation of lightweight, portable computing devices and high-speed wireless local-area networks has enabled users to remain connected while moving about inside buildings. This emerging paradigm has generated a lot of interest in applications and services that are a function of a mobile user's physical location. The goal here is to enable the user to interact effectively with his or her physical surroundings. Examples of such interactions include: printing a document on the nearest printer, locating a mobile user, displaying a map of the immediate surroundings, and guiding a user inside a building. As the surroundings change, so does the computing that happens. The interaction between computing and location may also be less direct. For instance, when in the boss' office, pre-fetch facts and figures on business performance and projects for ready access, but while in the cafeteria, turn on the sports score and stock ticker.

The granularity of location information needed varies from one application to another. For example, locating a nearby printer requires fairly coarse-grained location information whereas locating a book in a library would require fine-grained information [1]. In general, the amount of precision desired dictates the cost and complexity of the location-determination system.

To the best of our knowledge, previous research on in-building location systems has generally relied on specialized hardware and technologies that suffer from significant limitations and/or require extensive deployment of infrastructure solely for locating users. As an example, systems that use infrared (IR) wireless technology have been reported in [2], [3], [4], [5], and [6]. The limited range of an IR network, which facilitates user location, is a handicap in providing ubiquitous coverage. To overcome this problem, a few researchers have developed RF-tag based location systems as well [7], [8]. Unfortunately, these systems, like their IR counterparts, are often built for the sole purpose of determining users location, i.e., they do not provide any data networking services. Furthermore, the specialized hardware that these systems require are often cost prohibitive. The trade-off between deployment cost and perceived value of these systems has not been compelling enough for their large-scale adoption.

We have developed a system, called RADAR, which avoids many of the limitations of previous systems. Ours is a software-only system built over an off-the-shelf RF wireless local area network. A wireless LAN is typically deployed to provide tether-less data networking capability to mobile hosts. The location-aware services enabled by RADAR complement this already useful data networking capability of RF wireless LANs. This makes a wireless LAN more valuable and, in our opinion, increases the chances of large-scale deployment.

In a previous paper [10], we presented the design of the basic RADAR system and performance results obtained from a limited deployment of the system. Subsequent to that paper, we have expanded the deployment of our system using new wireless networking hardware and a different operating system platform. Over the past year we have gained substantial experience with the system, which has led us to add significant new enhancements to it and has given us insights into the inadequacy of the support for wireless networks in contemporary operating systems. This paper makes the following contributions:

1. We identify the shortcomings of the basic RADAR system that impact its deployability. We discuss three novel algorithms that we have designed and incorporated into RADAR to address these shortcomings. These include:
    a. An access point[1]-based *environmental profiling* algorithm that takes into account the reality that RF signals are significantly impacted by changes in the environment (e.g., change in number of people and obstructions in the building, change in temperature, etc [11]). Our algorithm enables RADAR to operate effectively even in the presence of significant shifts in RF propagation environment.
    b. A *history-based algorithm*, akin to the classical Viterbi algorithm [24], for continuous user tracking. This algorithm models physical constraints on a user's movements. In the process, it significantly diminishes error in user location estimation by disambiguating between candidate user locations guessed by the basic system.
    c. A *channel-switching algorithm* that enables RADAR to operate in wireless networks that employ frequency reuse techniques.
2. We identify shortcomings in the way contemporary operating systems abstract the wireless network hardware. We discuss this issue in the context of RADAR and present a sketch of WiLIB, a software library that we have developed. WiLIB exposes to the user-level aspects of the underlying wireless networking hardware that enable interesting new applications.

[1] An access point is a bridge between the wired and wireless networks.

The rest of this paper is organized as follows. In Section 2, we survey related work in the field of location determination. In Section 3, we describe the basic RADAR system. In Section 4, we describe our experimental testbeds. In Section 5, we discuss the performance of the basic system. In Section 6, we focus on the novel algorithms mentioned above and evaluate their performance. In Section 7, we present WiLIB. Finally, we present our conclusions in Section 8 and outline future work in Section 9.

2 Related Work

Related work in the area of location and tracking system falls into the following four broad categories: (1) IR-based systems (2) indoor RF-based systems (3) wide-area cellular-based systems, and (4) everything-else, e.g. ultrasound, magnetic fields, etc.

The seminal work in IR-based location systems is the *Active Badge* system reported in [2], [6]. In this system, a badge worn by a person emits a unique IR signal every 10 seconds. Sensors placed at known positions within a building pick up the unique identifiers and relay these to the location manager software. While this system provides accurate location information, it suffers from several drawbacks: (a) it scales poorly due to the limited range of IR, (b) it incurs significant installation and maintenance costs, and (c) it performs poorly in the presence of direct sunlight, which is likely to be a problem in rooms with windows.

Another system based on IR technology is described in [3]. IR transmitters are attached to the ceiling at known positions in the building. An optical sensor on a head-mounted unit senses the IR beacons, which enables the system software to determine the user's location. This system suffers from similar drawbacks as the Active Badge system.

An RF-based location-determination system called the *Duress Alarm Location System* (DALS) [7]. This system uses RF signal strengths to determine user location in a manner similar to our basic system. However, this system differs significantly from our enhanced system, which we describe in this paper. Also, DALS is different from our basic system in that it (a) depends on specialized hardware (b) requires infrastructure deployment over and above a wireless data network, (c) does not take into consideration the effect of the user's body orientation on RF signals, which our study shows can be significant, and (d) does not take RF propagation into account.

Another interesting indoor RF system is the 3D-iD RF tag system built by PinPoint Corporation [8]. Antennas planted around a facility emit RF signals at 2.4 GHz. Tags, acting like RF mirrors, transmit a response signal at 5.8 GHz along with an identification code. Various antennas receive the signal, and send the results to cell controllers, which triangulate the reflections to determine the tag's whereabouts. The system's locating ability varies depending on the number of antennas installed in an area but the best advertised resolution is 10 feet. The cost of an entire system is quite high. Once again Pinpoint's system differs from our system in that (a) it requires specialized hardware to do location determination, (b) they use signal-processing techniques that are significantly different from ours, and (c) their system does not include high-speed data networking capability.

The Daedalus project [17], a briefly mentions a wireless LAN based system for location estimation. This is a very coarse-grained user location system. Access points (APs) transmit beacons augmented with their physical coordinates. A mobile host estimates its location to be the same as that of the AP to which it is attached. Consequently, the accuracy of the system is limited by the (possibly large) cell size

In the wide-area cellular arena, several location determination systems have recently been proposed [13]. The technological alternatives for locating cellular telephones involve measuring the signal attenuation, the angle of arrival (AOA), and/or the time difference of arrival (TDOA). While these systems have been found to be promising in outdoor environments, their effectiveness in indoor environments is limited by the multiple reflections suffered by the RF signal, and the inability of off-the-shelf and inexpensive hardware to provide fine-grain time synchronization.

Systems based on the Global Positioning System (GPS) [12], [13], [14] while very useful outdoors, are ineffective indoors because buildings block GPS transmissions.

Researchers have also built systems using alternative technologies. One uses pulsed DC magnetic fields [15] to determine user orientation while another uses ultrasound signals [16] to determine user location. While these technologies and systems are very interesting, they generally suffer the same drawbacks as their IR and RF-tag counterparts. Their specialized hardware is generally targeted at niche markets, tending to make the system cost prohibitive, range limited, and unsuitable for large-scale deployment.

Our work differs from previous work in that we tackle the problem of people location and tracking using the widely available RF-based wireless LANs. With data networking speeds of up to 11 Mbps [18], wireless LANs have gained rapid acceptance and are widely being deployed in offices, schools, homes etc. Besides the existing wireless LAN our system does not require any additional hardware and can be enabled using purely software means.

These points are clarified in the following section.

3 The RADAR System

The RADAR system is built on a deployment of off-the-shelf wireless LAN technology. Access points (or base stations) are located in such a way as to provide overlapping coverage in the area of interest[2]. A mobile user carries with him/her a computing device equipped with a wireless LAN card capable of bi-directional communication with the access points.

The fundamental idea in RADAR is that in an RF network, the energy level or signal strength (*SS*) of a packet is a function of the receiver's (mobile user's) location. Consequently, it provides a means for inferring the user's location. There is a clear trend in SS as a user walks about the building. Not surprisingly, the signal received at the mobile is strongest when the receiver is close to the AP and weakest when it is far away. This strong trend, observed for all neighboring APs independently, is exploited by the system to estimate the mobile's location.

With this as motivation, RADAR takes the following approach to location determination. A *Radio Map* of the building is created. A Radio Map is a database of locations in the building and the signal strength of the beacons emanating from the APs as observed (or estimated) at those locations. So, for example, an entry in the Radio Map may look like $(x, y, z, ss_{i\ (i\ =1..n)})$ where (x, y, z) are the physical coordinates of the location where the signal is recorded and ss_i is the signal strength of the beacon signal emanating from the i^{th} AP.

Much of the effort in deploying RADAR goes into creating the Radio Map of the building. We evaluated two approaches for this purpose.

The *empirical* method for creating a Radio Map involves a mobile user walking to several different locations in the building and recording the physical coordinates of each location (e.g., using a floor layout map as reference) together with the signal strength of the beacon packets from each of the APs within range.

The *mathematical* method for constructing a Radio Map involves computing the received signal strength using a mathematical model of indoor RF signal propagation. We have developed a simple yet fairly accurate model that accommodates different building layouts while taking into account both free-space path loss and attenuation due to obstructions (e.g., walls) between the AP and the mobile.

To locate the position of the mobile user in real-time, the mobile measures the signal strength of each of the APs within range. It then searches through the Radio Map database to determine the signal strength tuple that best matches the signal strengths it has measured. The system estimates the location associated with the best-matching signal strength tuple, to be the location of the mobile.

[2] Having a network that is designed to provide overlapping coverage has an added bonus as it improves the system's performance and adds protection against downtime in the event of AP failure [23]

In case privacy is a concern, the architecture of RADAR enables a mobile to track its own location silently without other nodes in the system being aware of it. In the extreme, a mobile can essentially turn off data connectivity and use its wireless interface (in conjunction with RADAR) solely for the purpose of tracking its own location. Other than the SS values derived from beacons, the mobile only needs the Radio Map and the layout map of the building, which it can download say the first time it enters the building.

The specific search technique we developed is called *nearest neighbor(s) in signal space (NNSS).* The *NNSS* algorithm computes the *Euclidean distance* (in signal space) between each SS tuple in the Radio Map (ss_1,ss_2,ss_3) and the measured *SS* tuple (ss'_1,ss'_2,ss'_3). It then picks the SS tuple that minimizes the distance in signal space and declares the corresponding physical coordinates as its estimate of the user's location.

One variant of the basic NNSS algorithm is *NNSS-AVG*. The intuition here is that in case there is more than one SS tuple in the Radio Map that is "close" to measured SS tuple, there is little reason to pick just the closest one and discard others that are almost as close. So the NNSS-AVG algorithm picks a small number of closely matching tuples and averages their physical coordinates to obtain an estimate of the user's location. Often this composite estimate of location is more accurate than any of the individual estimates.

Of the two approaches to building a Radio Map, the empirical method performs better than the mathematical method. The *error distance*, which we define to be the Euclidean (physical) distance between the true location and the estimated location of the user, has a median of 2 to 3 meters, about the size of a typical office room. Unless otherwise indicated, the experimental results presented in this paper correspond to an empirically constructed Radio Map.

A more detailed discussion of the basic RADAR system and its performance appears in [10]. Subsequent to that paper, we have deployed a second RADAR testbed. There are significant differences in the hardware and technology of the two systems. In the next section, we describe both our testbeds, which we used for the experiments reported in this paper.

4 The RADAR Testbeds

In this section we describe in detail two significantly different deployments of RADAR based on different wireless hardware.

Our first testbed, which we used to build the basic system, is deployed on the second floor of a 3-storey building. The dimensions of the floor are 43.5 m by 22.5 m, an area of 980 sq. m (10500 sq. ft.), which includes more than 50 rooms. Three APs cover the entire floor. Each AP and mobile host is equipped with a Digital RoamAbout™ network interface card, which is based on Lucent's 2 Mbps proprietary WaveLAN™ RF LAN technology. The APs are attached to a Pentium-based PCs running FreeBSD 3.0 while the mobile hosts are Pentium-based laptop computers running Microsoft Windows 95. The network operates in the 2.4 GHz license-free ISM (Industrial, Scientific and Medical) band and has a range of 200 m, 50 m, and 25 m, respectively, for open, semi-open, and closed office environments [22].

Our second (newer) testbed is deployed on the second floor of a 4-storey building. The layout of the floor and the placement of the APs are shown in Figure 1. Five wall-mounted APs provide overlapping coverage in the portion of the floor where the experiments were carried out. In contrast to the first testbed, the new testbed is built over a standards-based state-of-the-art wireless LAN from Aironet Communications Inc. Specifically, we use Aironet's 4800 series of products, which includes the AP4800™ APs and the PC4800™ wireless network interface cards [23]. Like RoamAbout™, this RF hardware also operates in the 2.4 GHz ISM band. However, it has a

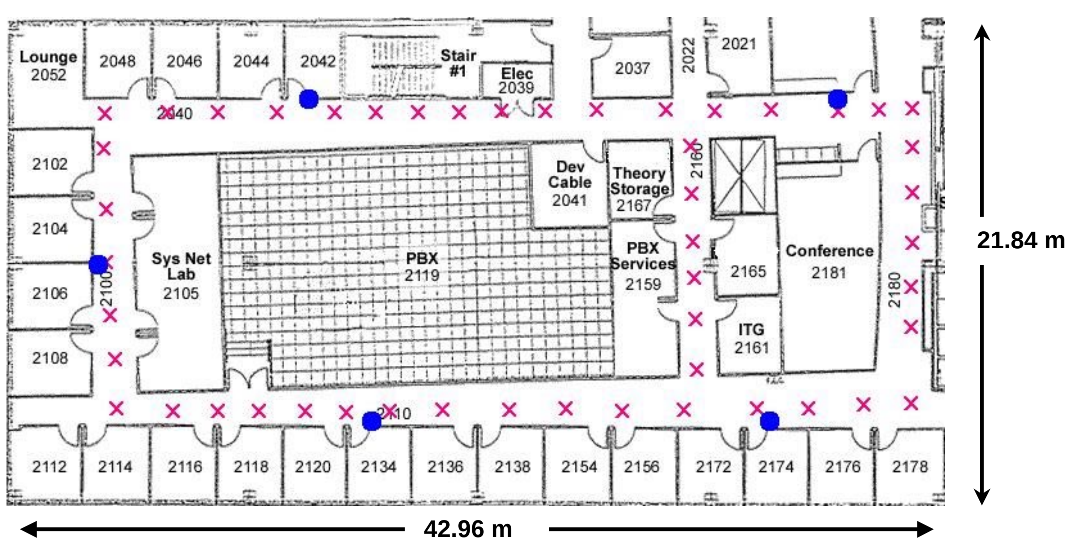

Figure 1: Map of the Aironet testbed. The crosses denote the locations were signal strength from beacon packets was recorded. The filled dots show the locations of the 5 access points.

different medium access control (MAC) and different physical (PHY) Layer. The 4800 is a multi-rate, direct sequence spread spectrum, IEEE 802.11b network [9]. It supports raw data rates of 1, 2, 5.5 and 11 Mbps and power levels of 5, 20, 50 and 100mW. The range of the network depends on the power-level and the data rate at which it is operating. Table 1 compares the two deployments of our system.

Table 1: Highlights of our testbeds

	Testbed 1	**Testbed 2**
Hardware	Digital Equip. Corp. RoamAbout™ (WaveLAN™)	Aironet Wireless Inc 4800 series
MAC	CSMA/CA [22]	IEEE 802.11b [23]
Modulation	Spread-spectrum DQPSK	Spread-spectrum CCK
Output Power	50 mW	30 mW
Data Rate	2 Mbps	1,2,5.5, 11 Mbps
Number of APs	3	5
Floor Dimensions	43.2 m x 22.5 m	42.9 m x 21.8 m
Number of points in Radio Map	70	49
OS platform	FreeBSD 3.0	Windows 2000

4.1 Summary of Previous Results

We briefly summarize the key results reported in [10]. These correspond to experiments conducted on the WaveLAN testbed.

1. The median error distance for NNSS is 2.65 m and 4.3 m with an empirically constructed Radio Map and a mathematically constructed Radio Map, respectively.
2. For our 980 sq. m (10500 sq. ft.) floor, the accuracy of location estimation reaches an asymptote when the Radio Map contains around 40 points.
3. The received signal strength tends to fluctuate even when the mobile is stationary. However, the accuracy of location estimation reaches an asymptote when the SS is averaged over 3 consecutive samples, which indicates that there is not much of a time lag in location estimation.

In the next, section we present some new results on the performance of the basic RADAR system, with particular focus on the new Aironet testbed.

5 Basic System Performance

We evaluate the base performance of RADAR by feeding the signal strength tuples recorded at known locations of the user into the NNSS algorithm (Section 3) and comparing the guessed location with the true location. This experiment simulates the case where we are trying to locate a static user. We quantify performance using the error distance. For much of our discussion, we focus on a single floor of the building, so the Euclidean distance is computed in two dimensions. (We discuss the effect of multiple floors in Section 6.3)

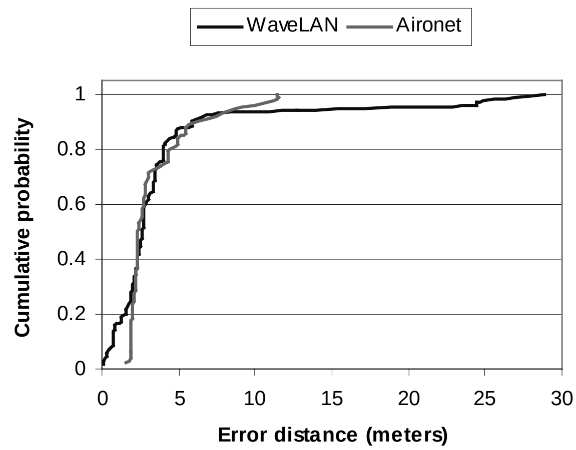

Figure 2: CDF of the Error Distance

Figure 2 plots the cumulative distribution function (CDF) of the error distance for the two deployments of our system. Since the WaveLAN deployment only had 3 APs, we consider only 3 APs for the Aironet deployment too for comparison purposes. We observe that the CDFs in both cases match well for the most part. The median error distance and the 90th percentile of the error distance for WaveLAN™ are 2.65 m and 5.93 m, respectively, while the corresponding values for Aironet are 2.37 m and 5.97 m. However, the tail of the CDF is much longer for WaveLAN™, implying that there are instances where the error distance is very large.

The reason for the long tail in the WaveLAN deployment is a phenomenon we term as *signal aliasing*. Two points that are far apart physically may be close together in signal space. Such aliasing can happen because of the complex indoor propagation environment. The signal strength at a point close to an AP may be similar to that at another point that is far away simply because of an obstruction (such as a wall) attenuates the signal received at the former point while the latter point receives an unobstructed signal. Whether aliasing occurs and how commonplace it is essentially a function of the building layout and the placement of APs. We discuss a novel technique to alleviate the effect of aliasing in Section 6.1.

5.1 Effect of the Number of Access Points

The larger number of APs with overlapping coverage in our new Aironet testbed (compared to the WaveLAN testbed) enabled us to investigate how the accuracy of RADAR is affected by the number of APs. This is an important question since APs contribute to the infrastructure cost.

The number of APs determines the dimension of the signal strength tuples that the NNSS algorithm operates on. A larger number of APs with overlapping coverage may make the NNSS search more accurate albeit at the cost of a larger hardware deployment. To quantify the benefit, if any, of increasing the number of APs, we varied their number from 1 through 5 in the Aironet testbed. The mean, median, and 90th percentile of the error distance are plotted in Figure 3.

The main observation is that while there is a significant benefit in going from 1 AP to 2 APs and again from 2 APs to 3 APs, there is little benefit in going beyond 3 APs. The inherent noise in the signal strength imposes a limit on how accurately location can be inferred using the NNSS technique, no matter how many APs provide coverage in a region. For instance, the signal strength can vary by a few dBm even while the receiver is stationary. The asymptote being reached at 3 APs indicates that RADAR does not require an excessive deployment of APs to function well.

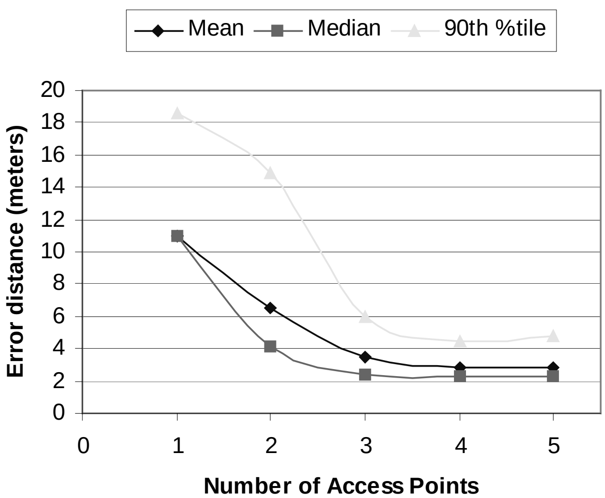

Figure 3: Impact of the number of APs on the error distance.

6 Enhancements to the Basic System

In this section we discuss some of the shortcomings of the basic system, describe the enhancements we have made to overcome these shortcomings, and present a performance evaluation of these enhancements.

6.1 Continuous User Tracking

The analysis in Section 5.1 focused on taking a static snapshot of the strength of the signals from multiple APs recorded at a mobile host and using this information to guess the location of the static user. The NNSS algorithm used for this purpose does not consider location information (or, to be more precise, guesses of user location) from the past.

The idea behind continuous user tracking is precisely to use information from the past to obtain a better estimate of a user's location. The intuition is that since physical constraints preclude a user from "jumping about" across large distances at random, the user's location at a given time instant is likely to be near that at the previous time instant. So, by tracking the user continuously, we complement signal strength information with the physical contiguity constraint to potentially improve the accuracy of location determination.

A beneficial side effect of continuous user tracking is that the problem of aliasing (Section 5) may be alleviated. Suppose that two physically distant points, A and B, are so close together in signal space (due to aliasing) that the NNSS algorithm is unable to disambiguate between the two. If RADAR was able to determine the location of the user unambiguously a little while earlier, it can pick between guesses A and B by using the unambiguous guess from the recent past in conjunction with the physical contiguity constraint.

The physical contiguity constraint has been employed elsewhere in the context of wireless networks. An example is the determination of a user's trajectory (for instance, while he/she is driving down a highway) to enable anticipation of handoffs in cellular telephone networks [20].

6.1.1 History-based Algorithm

Our history-based continuous user-tracking algorithm operates as follows. Each time a signal strength tuple is obtained by the mobile host, an NNSS search is done to determine the *k* nearest neighbors in signal space (*k*-NNSS), i.e., the *k* best guesses of the user's location. A history of depth *h* of such *k*-NNSS sets is maintained. The collection of these *h* *k*-NNSS sets can be viewed as a graph as depicted in Figure 4. There are edges only between vertices contained within consecutive sets. Each edge is assigned a weight to model the likelihood of the user transitioning (in successive time instances) between the locations represented by the two endpoints of the edge. The larger the weight is, the less likely is the transition. We use a very simple metric — the Euclidean distance between the two physical locations — as the weight.

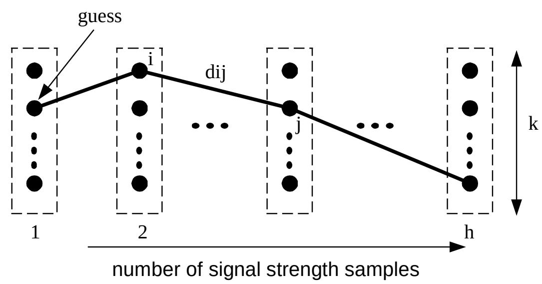

Figure 4: A depiction of the state maintained by history-based continuous user tracking algorithm. The shortest path is shown in bold. The location corresponding to the mid-point of the path is guessed to be the user's location. The weight of an edge between vertices i and j is d_{ij}, the Eucledian distance between the corresponding locations.

Each time the history vector is updated with the addition of the most recent *k*-NNSS set (and the deletion of the oldest set), the shortest path between the vertices in the oldest and the newest sets is computed. This shortest path can be viewed as representing the "most likely" trajectory of the mobile user. (This is similar to the Viterbi algorithm [24] in communication theory used by receivers to determine the most likely message to have been transmitted over a noisy channel.) Once the shortest path is determined, we guess the user's location to be the point at the start of the path (Figure 4). This procedure captures the physical contiguity constraint, but it also implies that there is a lag of *h* signal strength samples between when a user is at a location and when the system guesses the user's location.

6.1.2 Performance of History-based Algorithm

To evaluate the effectiveness of the history-based algorithm (HBA), we gathered signal strength data at 1-second intervals while the user was walking in the WaveLAN™ testbed. We also recorded the precise location of the user as a function of time. We then used the signal strength data from the walk in conjunction with the pre-computed Radio Map of the building to try and reconstruct the user's trajectory. We evaluated three different algorithms: NNSS, NNSS-AVG (where the physical coordinates of the 3 nearest neighbors in signal space are averaged to obtain an estimate of location), and HBA (with k = 3 and h = 6). The mean, median, and 90th percentile of the error distance are shown in Figure 5.

The main observation is that HBA significantly outperforms both NNSS and NNSS-AVG. The median error distance for NNSS (3.59 m) and NNSS-AVG (3.32 m) are 51% and 40% worse, respectively, compared to HBA (2.37 m). Also, the significant reduction in the 90% percentile of the error distance for HBA compared to the other two algorithms indicates that the long tail caused by aliasing (Section 5.1) has been shortened. This underscores the importance of tracking the user continuously.

In our analysis, we found that a history depth of h = 6 was optimal. This rather small history depth implies that the time lag caused by HBA is likely to be quite small. Since we collected an SS sample every second this translates into a 6-second lag. Although we did not do so in our experiments, it is entirely possible to reduce this lag by gathering SS samples more frequently.

Note that continuous tracking depends on signal strength samples obtained on a regular basis. This is clearly not a problem if the wireless interface is in active mode. But it is not a problem even if it is in power-saving mode. The interface will, in any case, have to wake up periodically to check if the AP has any messages queued for it. Signal strength samples can be conveniently obtained at such times.

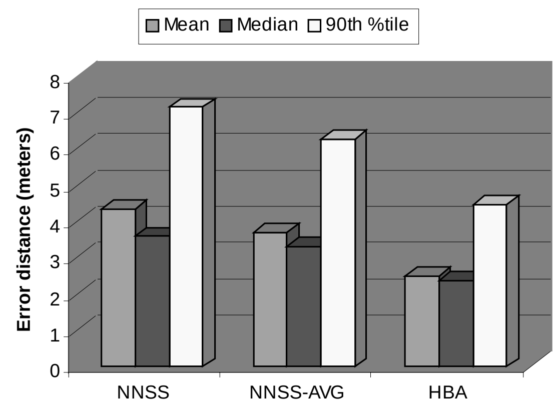

Figure 5: Performance of the various algorithms in tracking a user who is walking.

6.2 Profiling the Environment

In a previous section, we had mentioned that it might be necessary to construct a new Radio Map when the radio environment undergoes a shift. In this section we discuss this issue in greater depth.

6.2.1 Problem Description

Radio frequency presents a hostile environment for signal strength-based location systems. This is because signal propagation is dominated by reflections, diffraction, and scattering of radio waves caused by structures within the building [11]. The transmitted signal generally reaches the receiver via multiple paths (termed the *multipath* phenomenon). Multipath causes fluctuations in the received signal envelope and phase, and the signal components arriving from indirect and direct paths combine to produce a distorted version of the transmitted signal. Multipath within buildings is strongly influenced by the layout of the building, the construction material used, and the number of people in the building.

As the number of people in the building varies, the propagation characteristics of RF signals change as well. This is because the human body is made up of water, which absorbs RF signals. Our experiments show that a single human body may, on average, attenuate the signal by as much as 3.5 dBm.

As a day progresses, the number and distribution of people in a building may vary (due to meetings, mealtimes, etc.) causing the signal strength at various locations in the building to fluctuate considerably. Consequently, a Radio Map created at a particular time may not accurately reflect the environment at a different time. This can reduce the accuracy of the RADAR system considerably.

6.2.2 Solution

To account for the changes in the environment, we use multiple Radio Maps reflecting different environmental states. This brings up two questions: (1) how many Radio Maps should RADAR construct, and (2) how should RADAR dynamically pick the Radio Map that best represents the current radio environment.

We designed a novel solution to both of these problems by using the access points (APs) to calibrate the environment. The critical observation is that the APs are at fixed and known locations. Each AP listens for beacons (and other packets) from other APs within range and records the corresponding signal strength. The fixed location of the APs implies that any significant change in the signal strength is solely due to a shift in the radio environment. This provides a convenient means for the APs to determine when there has been a significant shift in the radio environment.

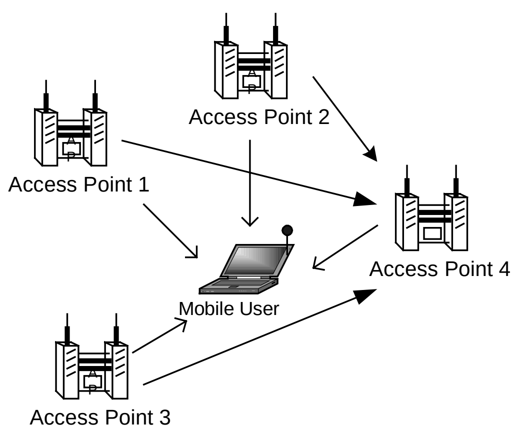

Figure 6: Access point-based environmental profiling: Beacon packets from neighboring APs are used to estimate (known) location of the target AP (AP4) using different Radio Maps.

The algorithm for determining how many Radio Maps to construct works as follows. RADAR starts off with an initial set of (one or more) Radio Maps. From time to time, each AP listens for beacons from other APs and uses RADAR to estimate its own location. (Figure 6 illustrates an example in which AP_4 determines its own location using AP_1, AP_2, and AP_3.) Knowledge of their true location enables the APs to compute the error in RADAR's location estimate. The APs repeat this computation using each of the Radio Maps in the pre-computed set and share the results with each other. A large error in each instance would indicate that none of the existing Radio Maps accurately represents the current radio environment, so a new map should be constructed to reflect the new environmental state.

While a similar algorithm could be used to answer the second question, viz. how RADAR should pick the most appropriate Radio Map dynamically, we use a more direct approach that avoids the need to repeatedly compute the error in RADAR's location estimate. As before, each AP records signal strength samples extracted from beacons and packets received from other APs within range. For each other AP, say AP_i, it computes the mean, m_i, of the received signal strength samples over a sliding window of w samples. It uses m_i together with the pre-computed mean (μ_e) and standard deviation (σ_e) of the signal strength corresponding to each environmental state, e, to estimate the likelihood that the received signal strength samples are in conformance with that environmental state e. We assume a Gaussian (Normal) distribution, $N(\mu_e,\sigma_e)$, for the signal strength and quantify the likelihood that the mean, m_i, conforms to the distribution using the probability density function (PDF) of the $N(\mu_e,\sigma_e)$ distribution. For each environmental state, e, the likelihood of match determined by each AP is multiplied together to obtain an overall estimate of the likelihood that the environmental state is e. The environment e_{max} with the highest likelihood of match is then guessed to be the true environmental state.

Each time a new set of signal strength samples is received (such as from periodic beacons), the sliding window used for averaging is moved forward one step and the computation described above is repeated. Whenever the computation determines a state other than the current one to be the best match, RADAR transitions to the new environmental state.

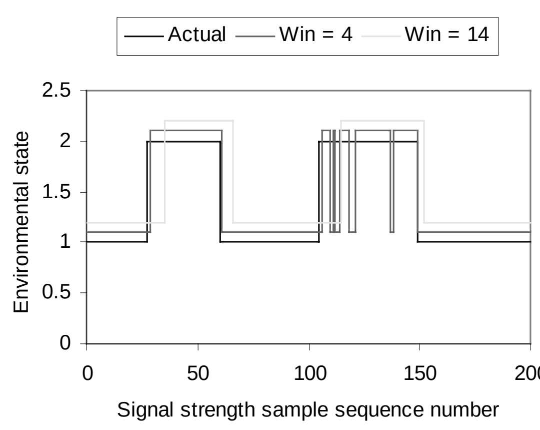

Figure 7: The transitions, both actual and inferred, between two different environmental states.

6.2.3 Experimental Validation

Our evaluation of environmental profiling is in two parts. First, we investigate the feasibility of having the APs probe the environment to track state changes (and accordingly pick the most appropriate Radio Map) accurately. Second, we evaluate the impact of environmental profiling on the accuracy of RADAR's location estimate.

6.2.3.1 Feasibility of Probing the Environment

We conducted a simple experiment to evaluate the effectiveness of our algorithm in tracking changes in the environmental state. We temporarily placed a pair of laptops in our campus cafeteria. We placed one laptop in a corner and ran a program that periodically broadcasts 4-byte UDP packets. We placed the second laptop in another corner and recorded the signal strength from broadcast packets. We performed the experiment during two periods in the day – one at lunch time, between 11:40 a.m. and 12:20 p.m. when there are many people in the cafeteria (*busy* period) and the other close to the end of the business day, between 4:00 and 4:40 p.m. when there are few people in the cafeteria (*lean* period). We used the first half of each trace to estimate the μ and σ for the corresponding environment. This information is summarized in Table 2. As we would expect, the greater concentration of crowds during the busy hour results in a smaller mean but larger variability in the signal strength.

	Mean (dBm)	Std. Deviation
Busy hour	46.07	2.41
Non-busy hour	50.05	1.19

Table 2: Characteristics of the received signal strength in two different environments.

We used the second half of the two traces to create an artificial signal strength trace by splicing together snippets from the two different environments alternately. This artificial trace had a total of 4 state transitions, as shown by the curve marked "Actual" in Figure 7. We show how well our environment state inference algorithm performs for two different sizes of the averaging window w — 4 and 14. For $w = 4$, there are several false transitions because the inference process maintains little history. On the other hand, with $w = 14$ the inferred transitions track the actual transitions well, but with a significant lag.

The accuracy of the inference algorithm is impacted both by the presence of false transitions and the time lag between the actual and the inferred transitions. We quantify both of these using a single metric, namely the number of time instances when the inferred state is different from the true state. We normalize the number of such errors by dividing by the number of actual state transitions in the duration of the artificial trace. This is plotted in Figure 8. We see that $w = 4$ minimizes the extent to which state inference is erroneous, despite suffering from a large number of false transitions.

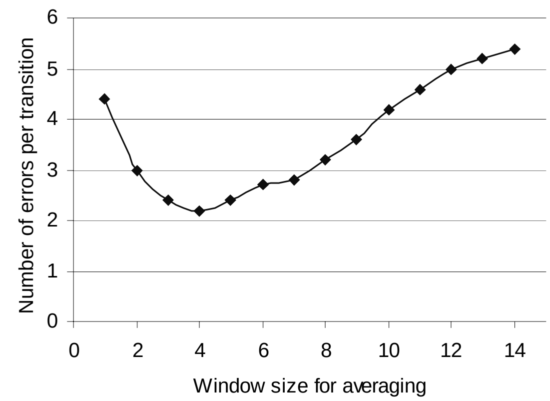

Figure 8: The error in inferring the environmental state as a function of the window size used for averaging signal strength samples.

In summary, this simple experiment suggests that it is indeed feasible to quickly infer significant changes in the environment using our inference algorithm.

6.2.3.2 Impact of Environmental Profiling on the Accuracy of Location Estimation

We now investigate how important it actually is to infer the correct environmental state and feed in the corresponding Radio Map to RADAR. In other words, does using the correct Radio Map have a significant impact on the accuracy of location determination?

Due to (non-technical) logistic constraints, we were unable to deploy a sufficient number of APs in the cafeteria, so we were constrained to conduct this experiment in our main office building, in which the Aironet network with 5 APs is deployed. This is a spacious and rather sparsely populated building. As such, there is not much variation in the RF environment over time. In contrast, we would expect a shopping mall or a restaurant to undergo significant shifts in the environment as crowds gather and dissipate. We simulated such variations in our environment by introducing artificial obstructions.

For the specific experiment described here, we constructed two different Radio Maps — one during normal operation (*lean* period) and another when 2 of the 5 APs had barriers placed right next to them (*busy* period). We took signal strength samples from the busy period and inferred user location using the NNSS algorithm. We did so in two settings — using the Radio Map constructed during the lean hour (this corresponds to environmental profiling *not* being performed) and using the map from the busy period (i.e., using the correct map determined via environmental profiling).

Figure 9 shows the results. Using the incorrect Radio Map results in far worse performance than when environmental profiling is used to pick the correct map. For instance, the 90th percentile of the error distance is 11.29 m in the former case compared to 3.16 meters in the latter case. This significant difference underscores the value of environmental profiling when there are significant shifts in the radio propagation environment.

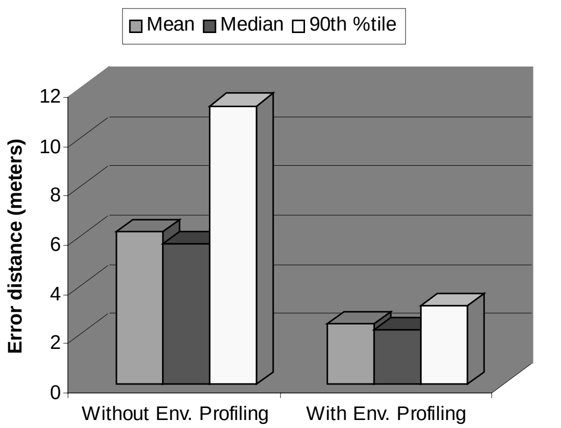

Figure 9: Performance of the NNSS algorithm with and without environmental profiling.

6.3 Effect of Multiple Floors

Our analysis thus far has been in the context of RADAR deployed on a single floor of a building. However, it is important to understand how RADAR would perform if deployed on multiple floors of a building. Specifically, signal aliasing between points on adjacent floors could cause RADAR to place the user on the wrong floor, which would be particularly undesirable.

To investigate this possibility, we conducted a limited experiment in the Aironet testbed. We picked 5 points on each of 3 contiguous floors in our building, with the points on each floor stacked right on top of the points on the floor below. This gave us a total of 15 points. We placed 3 APs on one of the floors and measured the beacon signal strength at each of the 15 points. We observed that the floor caused an attenuation of at least 9 to 10 dBm between points directly above or below one another. For points with different (x,y) coordinates on different floors, the attenuation was even greater. Our findings suggest that despite the physical proximity between points on adjacent floors, signal aliasing between such points is unlikely because the floor acts as a significant barrier to signal propagation. In the unlikely event of signal aliasing, our history-based continuous tracking algorithm (Section 6.1) will help negate the ill effects of aliasing.

Based on these observations, we conclude that RADAR would work well in a multi-floor environment, with APs deployed on one or more floors.

6.4 Effect of Multiple Channels

In designing our system we sidestepped one important issue that affects the deployment of RADAR. RADAR requires that the mobile node capture beacon packets from all the APs within range. To maximize system capacity, neighboring APs generally operate on different channels (a consequence of the classical frequency re-use requirement in cell-based networks [19]). So the mobile cannot hear beacons from all APs within range unless it switches channels.

As a first step, the mobile enters an *active scan* mode, in which it scans all channels to discover the identities of all APs within range[3]. In each channel it waits for a beacon packet emanating from the APs operating on that channel. The Aironet 4800 APs beacon every 100 milliseconds while the DEC's RoamAbout™ APs broadcast beacon packets every 200 milliseconds. Consequently, the waiting period could be long. For efficiency reasons we would like to minimize the mobile's overhead in switching to a channel, waiting for a beacon packet, and switching back.

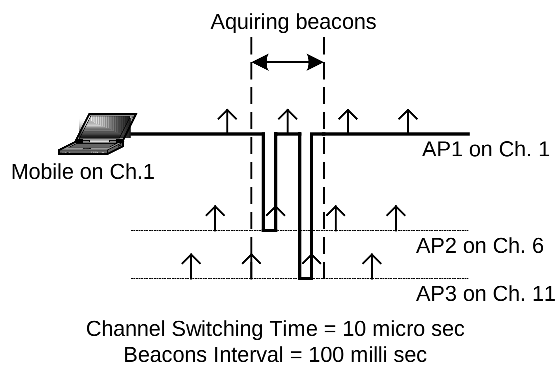

Figure 10: Mobile acquiring beacon packets from neighboring APs.

Our approach is to carefully schedule the channel switching so that minimal amount of time is spent waiting for beacon packets in the channel switched to. The idea is to synchronize the mobile with the APs and then exploit the fact that APs broadcast beacon packets periodically. If the mobile switches to a channel just before the beacon packet from the corresponding AP is expected, it minimizes the waiting period.

Figure 10 illustrates this concept graphically. A mobile operating on Channel 1, switches to Channel 2 and Channel 3 at appropriate times to grab the beacon packets. In between it switches back to Channel 1 to continue with its data communications. For our Aironet hardware we found that the channel switching time was a steady 10 microseconds. The time to grab the signal strength information from the kernel to the user-level RADAR daemon varies between 1 and 10 milliseconds depending on how loaded the system is.

[3] The 2.4 GHz ISM band is partitioned into 11 channels. Only 3 of these are non-overlapping. In our deployment we use only these three channels (Ch.1: 2412 Mhz, Ch 6: 2437 Mhz, and Ch. 11: 2462 MHz)

To estimate the tightness with which the mobile can be synchronized to the AP, we measured the round trip delay between the mobile and AP. We took two cases into account: (1) when the mobile is not transmitting any data to the AP, and (2) when the mobile is busy downloading streaming video over the network. We found that in the first case the round trip delay was about 3 milliseconds while in the second case it was about 15 milliseconds. Using NTP [25] we are able to synchronize nodes to within a few milliseconds, resulting in minimal overhead for channel switching. Packet loss during the short time slices in which the mobile has switched away from its primary channel are effectively masked by the link-level retransmission algorithm built into the IEEE 802.11 standard [9].

As discussed in Section 5.1, a mobile would ideally like to hear beacons from 3 or more APs to determine its location accurately. Since the mobile can move around and/or APs can fail, the number of APs within range that the mobile knows about can fall below the threshold of 3. At such times the mobile re-enters the active scan mode to update its list of APs within range. If the mobile is able to discover at least a threshold number of APs, it goes back to the optimized channel-switching mode discussed above. In the event that the mobile does not discover at least a threshold number of APs, it invokes the active scan mode periodically until it is able to hear from the desired number of APs. The periodicity of active scans could be a function of the volume of data communication that the mobile is currently engaged in.

7 Wireless is not Ethernet

Unfortunately current implementations of widely used operating systems have little or no support for local area wireless networking. Generally speaking, most OSs treat a wireless LAN as a wired LAN, exposing it to the higher layer networking protocols, operating system, and applications as just Ethernet. We believe that this is an incorrect paradigm that needs to be rectified.

Wireless networks are significantly different from traditional wired networks. These differences occur because of the continuously changing channel characteristics and host mobility. When operating systems and networking protocols ignore this fact, performance suffers and the potential for new functionality is lost. This last point is particularly relevant in the context of RADAR. The programming interface available to us did not provide the hooks needed to build a location-determination system such as RADAR.

RADAR requires signal strength measurements of beacon signals from neighboring APs to locate and track users. It requires knowledge of the identities of the APs within range and the ability to switch channels (frequencies) as described in the previous section. These requirements are specific to wireless networks and are not available as part of the programming interface for wired networks.

In FreeBSD, we found a WaveLAN device driver that provided at least some of the functionality we required (possibly an anonymous contribution from the research community). In the commercial Windows operating system we did not find the required functionality. Consequently, to extract the signal strength information from the AP beacons we extended Window's *Network Device Interface Specification (NDIS)* [21]. In particular, we added IoControl definitions in NDIS and corresponding functions in the hardware device driver to expose the wireless specific features of the underlying network.

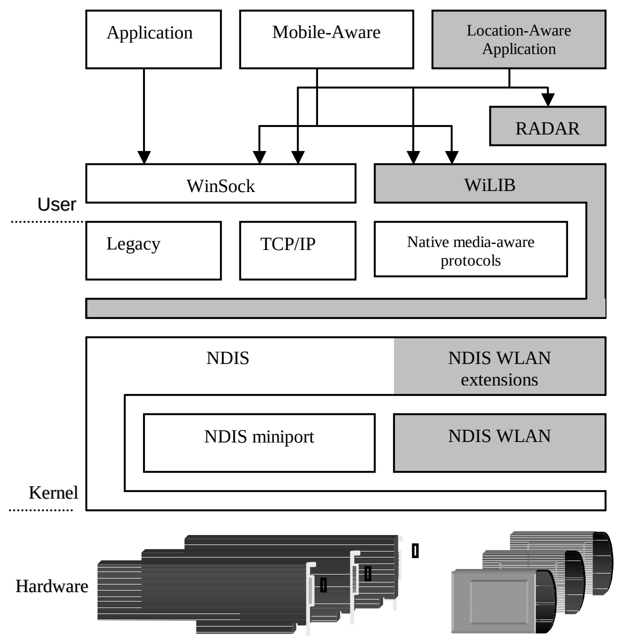

Figure 11: Placement of RADAR, WiLIB and our NDIS wireless extensions in relation to the networking protocol stack in Windows 2000. The gray regions correspond to modules that we have either added or enhanced.

In addition, we created a software library called WiLIB to provide application-level control of the wireless hardware (see Figure 11). Our high-level objective for WiLIB is two-fold: first, we want to enable the creation of novel user-level applications such as RADAR, and second, we want the ability to monitor and dynamically configure the hardware so that wirelessly connected systems can benefit from the latest research on adaptive algorithms that rely on knowledge of the state of the communications channel [26]. We are motivated by pushing WiLIB as an open software library for use by the research community to build wireless specific applications and algorithms that are agnostic of the wireless hardware and technology.

To create WiLIB, we have extended NDIS in three areas: *static queries, dynamic queries, and* *attribute setting commands*, all within the context of a generic wireless device. To avoid the need for polling, the ability to register callbacks for significant events (e.g., handoff to a new AP) is provided. To avoid overburdening the system with irrelevant processing, a mechanism for installing filters (e.g., MAC address based filters) in the driver is also provided. To enable these NDIS extensions we extended Aironet's NDIS mini-port driver. An NDIS mini-port driver is a hardware device driver that is generally provided by the hardware vendor. We enhanced the hardware device driver to expose functionality that could be used to address four areas: (a) performance (b) location awareness (c) energy management, and (d) mobility management. RADAR falls under the category of location awareness. The complete discussion of WiLIB is beyond the scope of this paper. However we point out that to enable RADAR, we incorporated the following functionality in WiLIB and NDIS:

(1) For each incoming packet from a particular MAC address, the application can retrieve its signal strength, noise floor at the transmitter, and noise floor at the receiver.
(2) The application can retrieve the list of all APs that the mobile can hear beacons from. This list includes all the above information for each of the APs plus their MAC addresses and beaconing frequency, and
(3) The application can configure the wireless NIC to operate at a specified channel. This allows for promiscuous mode of operation where the wireless NIC can be programmed to gather beacon information from all neighboring APs.

Beyond RADAR, WiLIB is allowing us to investigate other important areas that are specific to wireless networking such as:

- power and energy conservation (e.g., dynamically changing transmission power of packets)
- anticipatory hoarding of files [26] (e.g., by detecting reduction in signal strengths), and
- load balancing and mobility management (e.g., by intelligent management of handoffs).

Our experience in building RADAR has convinced us that operating systems need to specifically provide support for wireless network programming, as these networks have significantly different characteristics compared to their wired counterparts.

8 Conclusions

In this paper, we have described our experiences with RADAR, a software system we have developed for locating mobile users connected to a RF wireless LAN. The single most important contribution of RADAR is that it leverages the existing wireless data communications infrastructure to locate users rather than requiring a specialized hardware infrastructure.

Based on our experience with RADAR over the past year, we have identified limitations of the basic system and have presented novel algorithms to address these limitations. These include:

1. A history-based continuous user-tracking algorithm, akin to the classical Viterbi algorithm [24], which captures physical constraints on user motion. This algorithm alleviates the effects of signal aliasing and thins down the tail of the error distance CDF. Our experimental results indicate that in the absence of this algorithm, the median and the 90th percentile of the error distance degrade by over 50-60%.

2. An environment profiling technique, which makes the system resilient to variations in the radio propagation environment caused by factors such as crowds. The importance of this technique is apparent in the 3X degradation in location accuracy when environmental profiling is turned off and RADAR is forced to use a single Radio Map for widely different environments.

3. A channel-switching algorithm that enables RADAR to operate in wireless networks that employ frequency reuse techniques.

In the process of building the RADAR system, we encountered the shortcoming of contemporary operating systems in that they expose the wireless network interface to user-level applications just as if it were a wired network interface. To address this problem, we present WiLIB, a software library that exposes to the user-level facets of the underlying wireless networking hardware that enable interesting new applications.

Although for logistic reasons most of our experiments were confined to a single floor of a building, we did conduct a limited set of experiments on multiple floors of our building to confirm that signal aliasing is not an issue and that RADAR would work well in multi-floor buildings as well.

RADAR allows a mobile to track its own location in total privacy, if it so chooses. Once the mobile has downloaded the Radio Map and the layout map for a building, it does not need to communicate with the infrastructure (except for passively listening to beacons). We believe that this is a key strength of RADAR.

As a final note, we found that RADAR yields similar performance in our two testbeds despite significant differences in the floor layout and wireless network hardware. This gives us some confidence that our experimental results are not tied to a specific setting. That said, we realize that we need to experiment with RADAR in many more diverse environments before we would be in a position to draw any general conclusions.

9 Future Work

We are in the process of expanding our deployment of RADAR off-campus to a local shopping mall. Once this is in place, we plan to analyze the performance of RADAR in this very different setting.

Separately, we are considering the possibility of developing "light" APs. The sole purpose of a light AP would be to augment radio coverage in regions where the APs of the wireless LAN do not provide overlapping coverage. A light AP will only transmit beacons periodically and would not have any data networking capability, which would make it inexpensive. While a light AP does constitute infrastructure over and beyond that needed for data networking, it uses the same RF technology as the data networking thereby obviating the need for specialized hardware in the mobile hosts.

References

[1] G. J. Nelson, "Context-Aware and Location Systems," Ph.D. Theses, Cambridge University, U.K., January 1998

[2] R. Want, A. Hopper, V. Falcao, J. Gibbons. "The Active Badge Location System," *ACM Transactions on Information Systems*, Vol. 10, No. 1, January 1992, pp 91-102. (ORL Technical Report 92.1)

[3] R. Azuma, "Tracking Requirements for Augmented Reality," *Communications of the ACM*, Vol. 36, No. 7, pp: 50-51, July 1993

[4] N. Adams, R. Gold, B. N. Schilit, M. Tso, and R. Want. An Infrared Network for Mobile Computers. In *Proceedings of the USENIX Symposium on Mobile and Location-independent Computing*, pages 41-52, Cambridge, MA, August 1993. USENIX Association.

[5] F. Bennett and A. Harter. "Low Bandwidth Infra-Red Networks and Protocols for Mobile Communicating Devices," *Oracle Research Lab (ORL) Technical Report* 93.5.

[6] A. Ward, A. Jones, and A. Hopper. "A New Location Technique for the Active Office," *IEEE Personal Communications*, Vol. 4, No. 5, October 1997, pp 42-47. (ORL Technical Report 97.10)

[7] T. W. Christ and P. A. Godwin, "A Prison Guard Duress Alarm Location System", *Proc. IEEE International Carnahan Conference on Security Technology*, October 1993

[8] J. Werb and C. Lanzl, "Designing a Positioning system for Finding Things and People Indoors," *IEEE Spectrum*, (September 1998): 71-78 (also see http://www.pinpointco.com)

[9] IEEE Std. 802-11.1997, IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. Approved 26 June 1997.

[10] P. Bahl and V. N. Padmanabhan, "RADAR: An RF-Based In-Building User Location and Tracking System," Proc. *IEEE INFOCOM*, March 2000, http://www.research.microsoft.com/sn

[11] H. Hashemi, "The Indoor Radio Propagation Channel," *Proceedings of the IEEE*, Vol. 81, No. 7, pages 943-968 July 1993

[12] P. Enge, and P. Misra, "Special Issue on GPS: The Global positioning System," *Proceedings of the IEEE*, pp. 3-172, January 1999

[13] S. Tekinay, "Wireless Geolocation Systems and Services," Special Issue of the *IEEE Communications Magazine*, April 1998

[14] M. Moeglein, and N. Crasner, "An Introduction to SnapTrack^{TM} Server-Aided GPS Technology," available from: http://www.snaptrack.com/pdf/ion.pdf

[15] Ascension Technology Corporation, http://www.ascension-tech.com

[16] A. Ward, "Sensor-driven Computing," Ph. D. Thesis, Cambridge University, U.K., (May, 1999)

[17] T. D. Hodes, R. H. Katz, E. S. Schreiber, and L. Rowe, "Composable Ad Hoc Mobile Services for Universal Interaction," *MobiCom '97 Proceedings*, pp: 1-12, September 1997

[18] R. Nee, et. al., "New High-Rate Wireless LAN Standards," *IEEE Communications Magazine*, Vol. 37, no. 12, pp: 82-88, December 1999

[19] T. S. Rapport, Wireless Communications – Principles and Practice, *IEEE Press*, 1996

[20] T. Liu, P. Bahl, and I. Chlamtac, "Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks", *IEEE JSAC*, Vol. 16, No. 6, pp. 922-936, August 1998

[21] P G. Viscarola and W. A. Mason, *Windows NT Device Driver Development*, Open System Resources, 1999

[22] "Digital RoamAbout 915/2400 DS/PC Card and ISA Network Adapter: Installation and Configuration," *Digital Equipment Corporation*, April 1996

[23] "Developer's Reference Manual: PC4500/PC4800 PC card Wireless LAN Adapter," *Aironet Wireless Communications Inc.* 1999

[24] G. D. Forney, "The Viterbi Algorithm," *Proc. of the IEEE*, Vol. 61, March 1973, pp. 268-273

[25] D. L. Mills, "Network Time Protocol (Version 3): Specification, Implementation, and Analysis," RFC-1305, March 1992

[26] M. Satyanarayanan et. al., "The Coda File System," Carnegie Mellon University, Computer Science Department, http://www.coda.cs.cmu.edu/, 1994 – present