Spot Localization using PHY Layer Information

Souvik Sen
Duke University
souvik.sen@duke.edu

Božidar Radunović
Microsoft Research
bozidar@microsoft.com

Romit Roy Choudhury
Duke University
romit.rc@duke.edu

Tom Minka
Microsoft Research
minka@microsoft.com

ABSTRACT

This paper explores the viability of precise indoor localization using physical layer information in WiFi systems. We find evidence that channel responses from multiple OFDM subcarriers can be a promising location signature. While these signatures certainly vary over time and environmental mobility, we notice that their core structure preserves certain properties that are amenable to localization. We attempt to harness these opportunities through a functional system called *PinLoc*, implemented on off-the-shelf Intel 5300 cards. We evaluate the system in a busy engineering building, a crowded student center, a cafeteria, and at the Duke University museum, and demonstrate localization accuracies in the granularity of $1m$ x $1m$ boxes, called "*spots*". Results from 100 spots show that *PinLoc* is able to localize users to the correct spot with 89% mean accuracy, while incurring less than 6% false positives. We believe this is an important step forward, compared to the best indoor localization schemes of today, such as *Horus*.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Design, Experimentation, Performance

Keywords

Wireless, Localization, Cross-Layer, Application

1. INTRODUCTION

Precise indoor localization has been a long standing problem. While the frontier of localization technology has advanced over time, new kinds of location based applications are raising the bar. For instance, the advertising industry is beginning to expect location accuracies at the granularity of an aisle in a grocery shop [1]. Museums are expecting user locations at the granularity of paintings [2], so tourists can automatically receive information about the paintings they stop at. In addition to such high accuracy demands, these applications are inherently intolerant to small errors. If a localization scheme incorrectly places a user in the adjacent aisle in the grocery store, or displays information about the adjacent painting, the purpose of localization is entirely defeated. This is unlike traditional applications – say GPS based driving directions – where small errors are tolerable. As a consequence, new localization schemes will need to meet strict quality standards, without incurring a heavy cost of installation and maintenance. We refer to this problem as *spot localization*, where a device in a specific $1m$ x $1m$ box needs to be accurately identified. Localizing the device outside the box will be useless, irrespective of whether the estimated location is close or far away from the box.

The state of the art in indoor localization is quite sophisticated. Different schemes optimize distinct objectives, including accuracy [3–5], computation [4, 6], ease of calibration [7, 8], energy [9], etc. While the literature is rich, we sample few of the representative schemes to outline the frontier of today's location technology. Cricket [10] achieves high accuracy using special (ultrasound-based) infrastructure installed on ceilings. Noting the difficulties of installing special hardware, RADAR, Place Labs and Horus [4, 6, 8] explored the feasibility of using signal strengths from existing WiFi APs. While RADAR and Horus both rely on signal calibration, EZ [7] recently demonstrated the ability to eliminate calibration at the expense of accuracy degradation. Summarizing all these schemes, we find that the state of the art achieves median location error of $4m$ and $7m$, with and without calibration, respectively [7]. While this accuracy can enable a variety of applications, there are others that need precision at the granularity of "$1m$ x $1m$". This paper targets such high accuracies while ensuring that the calibration complexity is no worse than RADAR or Horus. We call our proposal *PinLoc*, as an acronym for Precise indoor Localization.

PinLoc's main idea is simple. While most WiFi based localization schemes operate with signal strength based information

at the MAC layer, we recognize the possibility of leveraging detailed physical (PHY) layer information. Briefly, the intuition is that the multipath signal components arrive at a given location with distinct values of phase and magnitude. When aggregated over multiple OFDM sub-carriers in 802.11 a/g/n, these rich data poses as a fingerprint of that location. Since we define each spot as a cluster of locations, war-driving each spot produces an array of location fingerprints. A training algorithm runs on each array of fingerprints to learn the statistical attributes of that spot. Later, when a mobile device arrives at a spot, it computes a fingerprint (from a sequence of overheard beacons), and classifies it to one of the spots by matching against the learnt attributes. We find that devices are reliably classified to the correct spot, despite movements of people and other objects in the environment. Our war-driving effort is comparable to RADAR or Horus – we mounted a laptop on a Roomba robot and programmed it to move randomly within each spot for around 4 minutes. Finally, where several other schemes are strongly reliant on multiple APs, PinLoc offers reasonable performance even in WiFi-sparse environments. In some cases, PinLoc is able to localize even with signals from a single AP.

At first glance, our findings seemed too good to be true. We expected the signal phases to be sensitive to the orientation of the laptop, human movements, and/or structural changes in the environment (such as repositioning of chairs, boxes, shelves). We suspected that frequent war-driving would be necessary to adapt to such environmental perturbations. While these concerns were natural, we were surprised to find that the fingerprints actually preserved statistical properties even under perturbations. For instance, although the channel response at a specific location varied with time and environmental dynamism, they could be consistently organized around a *set of few tight clusters*. When combined across 30 subcarriers and different APs (i.e., high-dimensional data), we found that even the sets of clusters could be reasonably unique. Further, since spots are composed of many "distinct locations", the fingerprint of a spot is a string of channel responses from multiple distinct locations inside that spot. Thus, even if the channel response from one distinct location is not unique, the probability that the string of channel responses appears in more than one spot is far lower. These and other factors (discussed later) together contribute to PinLoc's robustness. RSSI, on the other hand, is an average of the magnitudes on each sub-carrier, which hides fine-grained information about that location, ultimately limiting the accuracy of localization.

Harnessing the above opportunities into a working system (using off-the-shelf wireless cards) forms the core of PinLoc. The detailed PHY layer information is first extracted from the driver and sanitized using a phase correction operation. The sanitized parameters are then fed to a machine learning algorithm that models the channel response distribution. Later, during system tests, the channel parameters are extracted from received WiFi beacons, and classified to one of the war-driven spots. To address energy issues, PinLoc disables active scanning, and only uses beacons from APs in the same channel. Finally, the individual modules are combined into a full system, and tested over a variety of scenarios. The results are promising – with less than 4 minutes of wardriving per spot, we observe 89% mean accuracy and false positives consistently below 6%. From the application's perspective, Pin-

Loc was tested in the modern art wing of Duke University's museum. Spots in front of each of 10 paintings were localized with high accuracy.

To the best of our knowledge, no prior work has demonstrated PHY layer-based WiFi localization on off-the-shelf platforms. Zhang et. al. [11] used signal amplitudes and phases on USRP platforms to demonstrate *location distinction*. We note that location distinction detects when a node's location has changed (e.g., for security purposes), but does not need to establish uniqueness for each location. Localization is naturally a far stricter problem, especially when the target is sub-meter accuracies. PinLoc makes an early effort towards this goal – the main contributions may be summarized as follows.

- **We target the problem of** *spot localization* **where success is defined as the ability to place a device within a** $1m$ **x** $1m$ **area, called spots.** We break away from RSSI based schemes and explore the feasibility of using detailed PHY layer information.

- **We utilize the per-subcarrier frequency response as features of a location, and rely on machine learning algorithms to classify a device to one of the trained spots.** We use off-the-shelf Intel 5300 cards; the entire system relies on existing WiFi deployments, and requires no special installation.

- **We evaluate PinLoc at varying accuracy standards, namely, discriminating between seats in a lab, chairs in a cafeteria, and adjacent paintings in a museum.** We observe consistent accuracies under mobile/dynamic environments, outperforming Horus [4], the most accurate RSSI based localization.

The subsequent sections expand on each of these contributions, beginning with definition and applications, followed by measurement, design, and evaluation.

2. LOCATIONS, SPOTS, AND APPLICATIONS

The above section loosely used the terms "locations" and "spots" – we clearly define them here. Locations are like pixels that define the resolution of our localization system. Each location is a small area that has a unique fingerprint. As we show later, the "size" of a location is approximately $2cm \times 2cm$. Spots are larger boxes, say $1mx1m$, and composed of multiple locations. We will see later how the fingerprint of a spot is essentially a combination of fingerprints from all locations in that spot.

Apps for Spot Localization.
The ability to localize users to the granularity of a spot is obviously useful — for instance, precisely tracking a user's indoor location can empower numerous applications. What might be less obvious is that a reasonable number of applications may be enabled even if only a few spots are reliably identified. For instance: (1) Advertising agencies may post discounts on to the user's phone when she pauses in front of select products in the store. (2) Spot localization may be applicable to geo-ofencing – students at different desks may see different sets of exam questions. (3) Logical locations [12] refer to places (Starbucks, Airport, public library) as opposed to geographic coordinates (latitude/longitude from GPS). Since such places

are often adjacent to each other, separated only by a wall, it has been difficult to tell the exact place in which a person is located. Spot localization may be able to detect when a user enters/exits through a door, thereby identifying the place of the user. In a related application, a blind person could come to an approximate location and be prompted with "*the door in front of you is Dr. Brown's eye clinic*". This may also aid security applications, the content from a server may be downloadable only when a person is inside a restricted area [13]. (4) Even identifying which aisle a person is in could be achieved if the entry and exit points of an aisle are spot-localized. We argue that PinLoc is particularly amenable to these applications since the war-driving effort gets proportionally reduced with fewer spots. While the same reduction may apply to all war-driving based schemes in general, their error margins may still be an issue. Horus [4], for example, incurs $4m$ mean error, which may not discriminate between adjacent grocery aisles or between two adjacent wide-screen TVs at Best Buy. PinLoc is tasked to achieve this discrimination.

3. HYPOTHESES AND MEASUREMENTS

This section aims to show through experiments that PHY layer channel information from existing WiFi deployments can be an indicator of location. We identify the essential hypotheses that must hold, and verify them through measurements. The next section draws on the findings to design and implement the machine learning components in PinLoc. We begin our discussion with a brief background on channel frequency response.

3.1 Background

Most modern digital radios use OFDM communication, and transmit signals across orthogonal subcarriers at different frequencies. Each transmitted symbol $X(f)$ is modulated on a different subcarrier f, and the quality of the received symbol $Y(f)$ will depend on the channel $H(f)$:

$$Y(f) \quad = \quad H(f)X(f) \tag{1}$$

Vector $\mathbf{H} = (H(f))_{f=1,\cdots,F}$ is called *channel frequency response* (CFR), and it is a complex vector that describes the channel performance at each subcarrier. A 802.11 a/g/n receiver implements $F = 48$ data sub-carriers, and includes a channel estimation circuit as a part of the hardware. The Intel 5300 [14] card, released recently with a publicly downloadable driver, exposes the per-subcarrier CFR to the user. Figure 2(a,c) shows examples of some CFR vectors.

Two important properties of the CFR are of interest in PinLoc. (1) The CFR changes *entirely* once a transmitter or a receiver moves more than a fraction of a wavelength [15]. Since the WiFi wavelength is about $12\ cm$, the CFR offers a possibility to discriminate between two nearby locations. (2) Even when the device is static at a specific location, the CFR experiences *channel fading* due to changes in the environment at different time-scales. This introduces randomness in the CFRs, injecting ambiguity in signatures. However, it is unclear whether this randomness is completely unpredictable or whether it exhibits some statistical structure that lends itself to localization. The following hypotheses and measurements are designed to answer these and related questions.

3.2 Hypotheses

We present 3 main hypotheses that need to hold if CFRs are to be used for PinLoc.

1. *The CFRs at each location appear random, but actually exhibit a statistical structure.* This structure is preserved over time and environmental changes.

2. *The "size" of the location (over which the CFR structure is defined and preserved) is small.*

3. *The CFR structure of a given location is different from the structures of* all *other locations, with high probability.* The probability increases when aggregating the CFRs from multiple APs.

Towards verifying these hypotheses, we first describe our testbed environment and experiment methodology, followed by the measurements and findings.

3.3 Experiment Methodology

Our initial experiments were performed in a relatively busy engineering building (see floorplan in Figure 1). We consider a set of 15 different spots in our lab and the adjacent classroom. The center of these spots are approximately $2m$ apart from each other. We place a laptop equipped with the Intel 5300 WiFi card [14] at each of these spots, and associate them to existing WiFi APs (the same set of WiFi APs are audible from each of these spots). The laptops are made to download packets through each of the nearby APs (using iPerf) – the corresponding channel frequency responses (CFR) are recorded for each packet. The Intel 5300 card firmware exports a subset of the CFRs (30 out of 48 data subcarriers), and we only use these for our scheme.
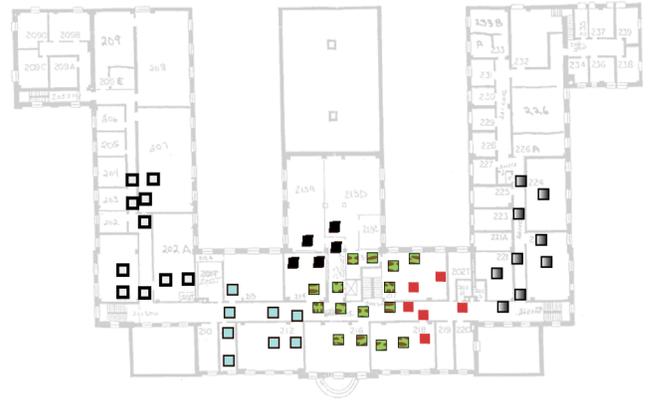
Figure 1: Engineering building floorplan. Different sets of spots shown in different colors – our initial measurements in this section uses only one set of 15 spots (shown in green).

.

For each location we perform 4-6 measurements at different times during busy office hours. During the measurements in the lab (a $10m$ x $10m$ area), there were between 3 to 5 people who frequently walked in and out. Classroom measurements were performed during and between classes (classroom capacity of 24 seats) – one measurement coincided with all students exiting the classroom at the end of a class.
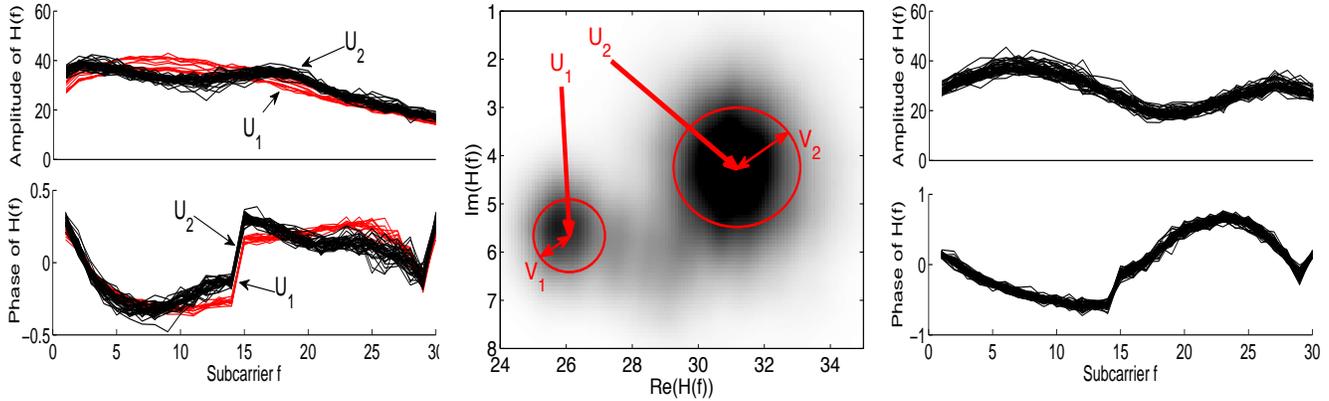
Figure 2: **(a) The amplitudes and phases of the channel responses H of 50 (out of 20000) packets sent over the same link (we see two unique clusters, U_1 and U_2); (b) PDF of the complex value of the same 20000 channel responses $H(f)$ for a single subcarrier $f = 20$; (c) The amplitudes and phases of the channel responses H of 50 packets at a different client location.**

3.4 Measurement and Verification

(Hypo. 1) *The CFRs at each location appear random but actually exhibit a statistical structure over time.*

Testing on a Single Location: Figure 2(a) shows the channel frequency responses (CFR) recorded on a laptop at a fixed location (the laptop received $20,000$ packets from a specific AP over a period of $100s$, but for visual clarity, we only show 50 CFRs from 50 randomly selected packets). We observe two emerging clusters, denoted with two vectors U_1 and U_2 – CFRs belonging to the same cluster are not identical but appear as noisy realizations of the cluster mean. This is an outcome of fading, caused by different electro-magnetic propagation effects and/or environmental changes.

We now take subcarrier $f = 20$, gather all its CFRs from all $20,000$ packets, and plot the empirical probability density function (PDF) in Figure 2(b). The CFRs are complex numbers, and hence we plot the *Real (Re)* and *Imaginary (Im)* values on X and Y axes – darker colors represent higher values of the PDF. We again see that two dominant clusters emerge, each cluster appearing as a complex Gaussian random variable, with means $U_1(f)$ and $U_2(f)$ and variances $V_1(f)$ and $V_2(f)$, respectively. Of course, this is only a visual indication – we will carefully model this later in section 4.2.

Figure 2(c) shows the outcome of the same experiment, but with the laptop placed at a different location. We find only a single cluster of CFRs, and the shapes of the CFRs differ distinctly from those in Figure 2(a). These few representative clusters hint at the possible existence of complex but invariant structures in per-location CFR, motivating further investigation. Figure 3 pictorially explains the cluster computation process.

Temporal stability of CFR Clusters: We now test whether the observations from these two locations generalize to a larger number of locations, under various environmental changes. Figure 4 shows the distribution of the number of representative CFR clusters from 30 distinct links in total. For each of the 30 links we aggregate all the available data, collected over 2 days – this adequately captures the links' temporal fluc-
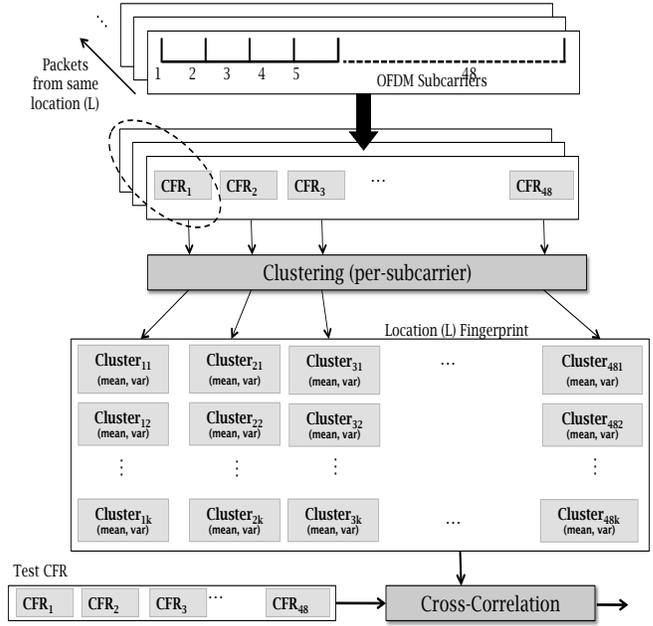
Figure 3: **The process of creating clusters (which together form the location fingerprint), and how test CFRs are cross-correlated with the fingerprint.**

tuations. We use the clustering algorithm, described later in section 4.3, to identify representative CFR clusters. Evident from Figure 4 (a), more than 80% of links experience 4 CFR clusters or less. However, we still see a substantial number of links with a large number of clusters, even up to 19. This could well suggest that the CFR structure is quite random in dynamic scenarios (e.g., in the classroom), and thus, PinLoc may only be applicable in very static environments.

To verify this, we next look at frequency of occurrence of different clusters. Figure 4(b) shows that the distribution is highly non-uniform, with a strong predominance of the more frequent clusters (i.e., the frequent clusters occur very frequently, and the vice versa). Evidently, the fourth-most frequent cluster occurs no more than 10% of the cases in any
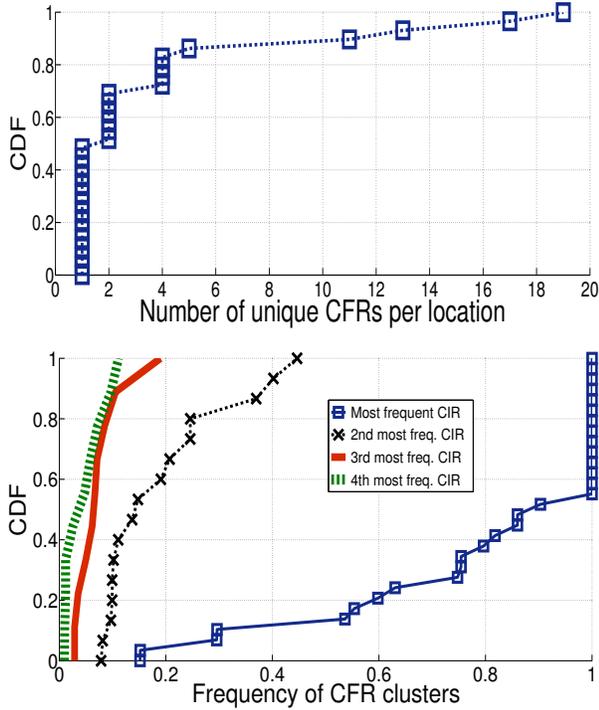
Figure 4: (a) CDF of the number of CFR clusters observed at 30 different client locations (i.e., 30 distinct links); (b) CDF of the probability of seeing the n-th most frequent CFR cluster.

spot, and the 5th, 6th, ... 19th clusters are almost rare. This suggests that even if a few spots experience large number of clusters, we are not very likely to see most of them, both during the training and localization.
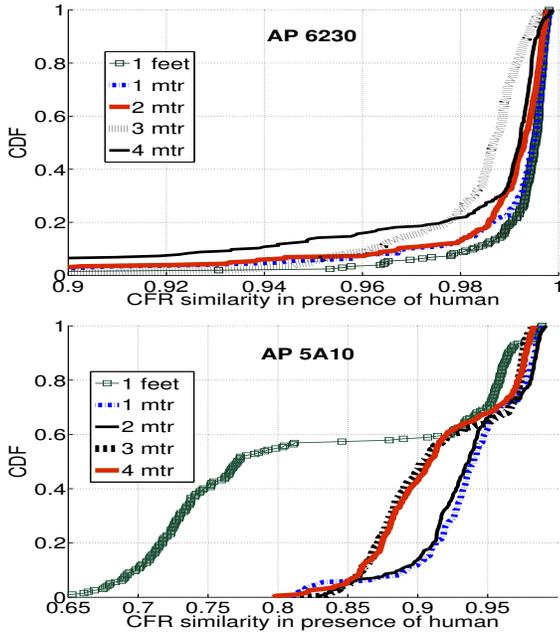
Figure 5: CFR cross correlation in presence of human beings at a location for 2 different APs at 2.4GHz

Impact of environmental changes: To understand the impact of environment changes on CFR clusters, we perform two controlled experiments. First, we study the effect of human mobility on CFR stability. We place a laptop at a fixed location and start gathering CFRs from two different APs. We run the experiment during night, and observe a single CFR cluster for both links. Then, we position a human at an increasing distance d from the laptop. We plot the CDF of cross-correlation of each received CFR with the CFR cluster observed without the human (Figure 3). We define cross-correlation[1] of two CFR cluster means $\mathbf{a}$ and $\mathbf{b}$ as

$$c(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=1}^{F} a_i b_i}{\sqrt{\sum_{i=1}^{F} a_i^2} \; \sqrt{\sum_{i=1}^{F} b_i^2}}. \qquad (2)$$

Figure 5(a) shows high correlation, suggesting that human obstructions may not create a significant change to the CFRs from $AP6230$. This is probably because the human does not alter any of the strong signal paths between the laptop and the AP. The link to $AP\ 5A10$, however, changes with human movement; nevertheless, the change is substantial only when the human is very close to the laptop (1 foot away). Even then the median cross-correlation is larger than 0.75. In all other cases, the cross-correlation remains high, suggesting that human movements around the device may not distort the CFRs too much from its core structure.
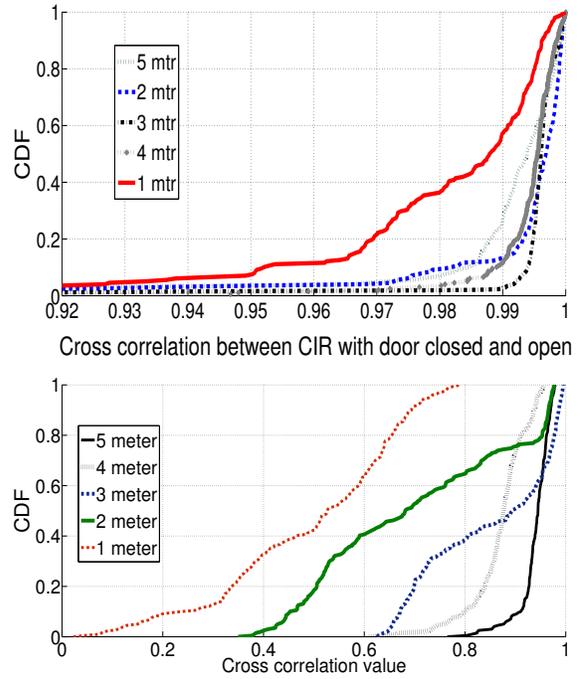
Figure 6: (a) CFR cross correlation for (a) door open vs. closed and (b) original metal shelf vs. moved; for various distances from the laptop.

Next, we study the effects of moving objects in the environment – doors, chairs, metal shelves, etc. We place a laptop at a fixed location and gather CFRs from different APs. Figure 6(a) shows that wooden door obstructions do not induce a significant change to the CFRs. Our measurements also

---

[1]We use cross-correlation as a metric here for illustrational purposes, and we design a more accurate metric in section 4.
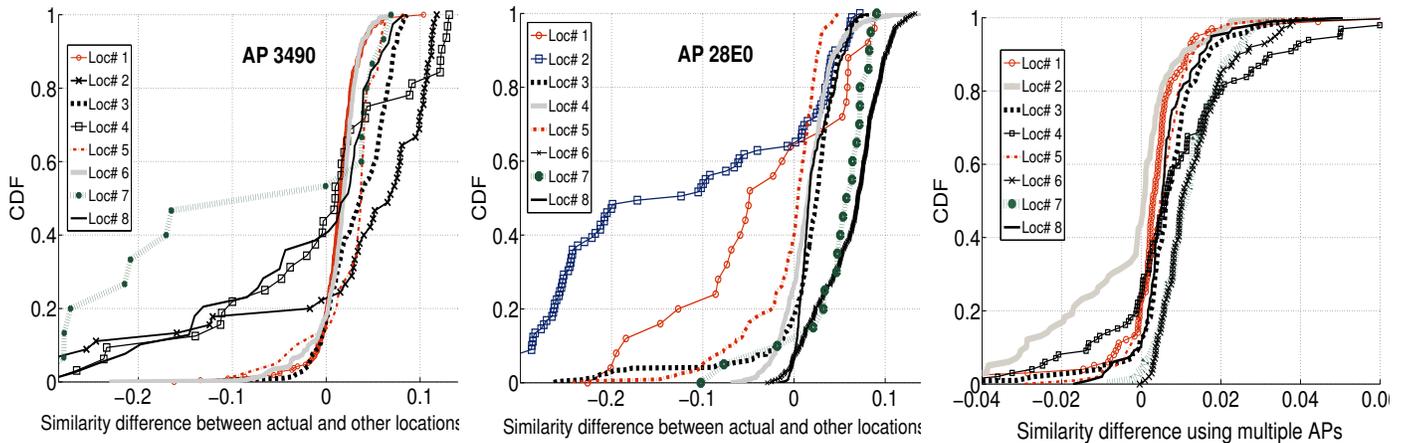
Figure 7: CDF of the difference in similarities $S_{own} - S_{others}$ **observed at 8 locations, for two different access point: (a) AP 3490, (b) AP 28E0. (c) CDF of the maximum similarity difference (** $S_{own} - S_{others}$ **) across all APs.**

show similar results for other non-metallic objects. However, a repositioned metallic cupboard (approximately 3.5 feet high and 3 feet wide) altered the CFRs significantly – Figure 6(b) shows the impact. Importantly, however, these alterations are localized only around the shelf's original and final locations; spots more than 4 meters away from the shelf are much less perturbed, and need not be re-calibrated. While the above results are from controlled experiments, section 5 reports results from uncontrolled settings (student center, cafe, museum), with hundreds of mobile humans and shifting objects.

(Hypo. 2) *The "size" of the location (over which the CFR structure is defined and stable) is small.*

Precise localization will require the CFR structure to vary over space. If the structure varies in large granularities (say, multiple meters), PinLoc's accuracy will naturally be bounded by that granularity.

Thus to understand the "size" of each location, we move a test location increasingly further away from a reference location, and compute their respective CFR cross-correlations. Various existing channel measurements [11,15] show that the channel changes entirely once a receiver is moved more than a wavelength, which is $12cm$ in the case of WiFi. Figure 8 shows that the cross-correlation drifts apart with increasing distance, and is quite low even above $2cm$. While this may suggest that localization is feasible at $2cm$ resolution, we will see later that multiple locations in a room may exhibit matching fingerprints. This is why we will require PinLoc to collect multiple fingerprints from very nearby locations (within a $1m \times 1m$ spot). The combined fingerprint from a specific "spot" is much less likely to occur in other spots, and hence, we will attain reliable localization at the granularity of spots.

(Hypo. 3) *The CFR structure of a given location is different from the structures of all other locations.*

To evaluate the (dis)similarities of CFRs among different locations, we divide the measured data into a training and a test set. Each location has a set of CFR clusters pertaining to an AP, represented by their mean and variance and learnt from the training set. For a test CFR from a location L, we use correlation to find the best matching CFR cluster from L's training data. The correlation value, denoted as S_{own}, indicates *similarity* of the test CFR with the trained fingerprint at
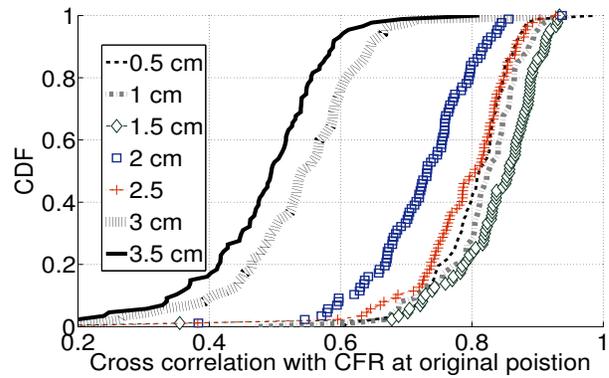
Figure 8: Cross-correlation drifts away for CFRs that are apart by 2cms or more.

the same location. Now, for all fingerprints of all other locations, we find the one that exhibits maximum *similarity* of this test CFR – we denote this similarity as S_{others}. If a device's measured CFR is more similar to a different location than its own, we will naturally misclassify the device's location.

Figure 7 (a) and (b) plot the CDF of the difference in similarities, $S_{own} - S_{others}$, for 8 different locations, for two distinct APs[2]. If the difference is negative, then the packet is likely to be misclassified. Also, the larger the difference, the greater the confidence in packet location. Figure 7 (a) and (b) show that the CFR *from a single AP* is often sufficient to correctly classify location. Of course, in some cases – such as ($Location$ 7, $AP3490$) – more than 50% of the CFRs are more similar to other locations, implying misclassification. However, when considering the CFRs of location 7 to a different AP, the misclassification reduces significantly. This suggests that CFRs are diverse across different APs, and this diversity can be leveraged to improve localization. Figure 7 (c) shows the effect of exploiting AP diversity with 2 APs. Specifically, we now pick the AP with the highest similarity difference. Clearly, there is significantly less negative values in Figure 7 (c), implying that AP diversity can help create dissimilarity in location fingerprints.

---

[2]Note that S_{own} and S_{others} are computed per-AP.

One may ask: *Figure 7 shows that a packet may be classified to one out of 8 different spots. In reality, the system will need to discriminate between many more spots – will the system scale to such scenarios?* We note that PinLoc does not need to discriminate between all spots in a large area. Prior work has used WiFi SSIDs alone to localize devices to around $10m$ x $10m$ regions in indoor environments [8]. PinLoc will leverage such schemes to first compute a coarse-grained macro-location, and then discriminate only between the spots inside that macro-location. Having verified these hypotheses, we now design the full localization scheme, including CFR clustering and matching over multiple sub-carriers. Thereafter, we evaluate PinLoc's performance in section 5.

4. DESIGN AND IMPLEMENTATION

Figure 9 shows the architectural overview of PinLoc. During war-driving, a Roomba-mounted laptop moves randomly through a spot for around 4 minutes. Recall from sections 1 and 2, that a spot is composed of many "locations" (Figure 10), hence, the laptop measures the CFR of every location it hits. Of course, due to the Roomba's random mobility, the laptop may not be able to collect CFRs from all locations in a spot, making our war-driving process far less painstaking. The war-driven data is then sanitized through a phase correction operation and fed to a clustering algorithm, which outputs a few dominating CFR clusters per-location. As mentioned earlier, these clusters are expressed as cluster means and variances, and together form the training set. The training set from different locations in the same spot are stored under the corresponding spot database.
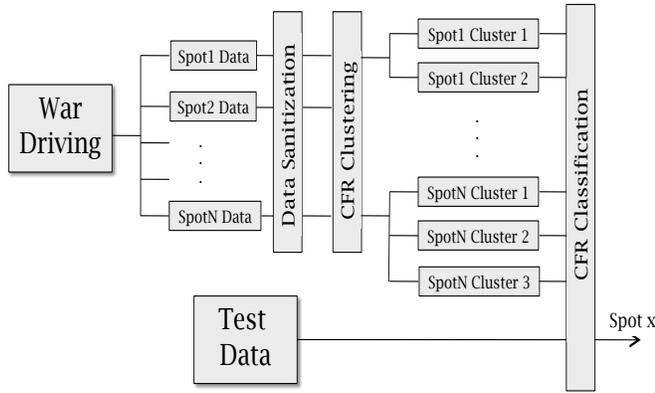
Figure 9: PinLoc architecture

During the real-time localization phase, each mobile node passively records *strings* of CFRs it receives from AP beacons (string length of 4 denoted with shaded squares in Figure 10). The mobile either sends these CFRs to a PinLoc server, or requests the spot databases for candidate spots in its (known) macro-location. The next step is matching. A single CFR reading will not match exactly the ones from the spot-databases due to random fluctuations, but on average, they are more likely to fall in the correct cluster than a wrong one. To improve the accuracy, instead of matching a single reading, we match the string of consecutive CFR readings. Each of these 4 CFRs may match well with a location from a random incorrect spot, but it is unlikely that all the CFRs from a string will match better with locations from the same wrong spot.
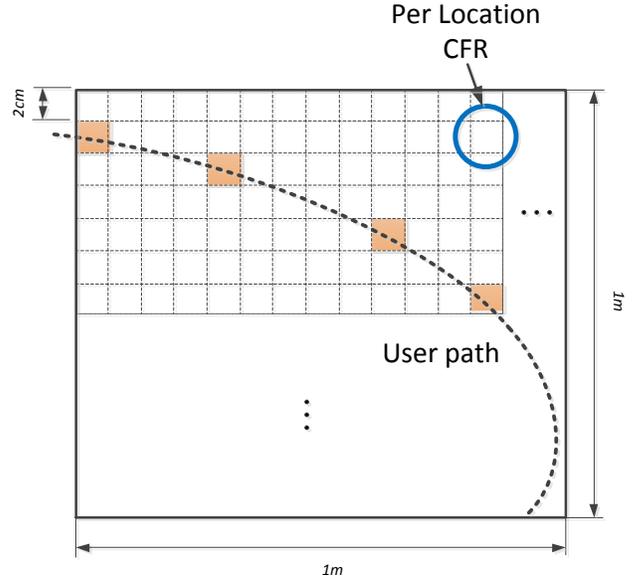
Figure 10: A device records multiple CFRs from a spot.

Results from the next section will confirm this robustness of spot localization. However, before presenting the results, we discuss the sanitizing, clustering, and matching modules in details.

4.1 Data Sanitization (Phase and Time Lag)

The CFRs received at a location cannot be directly used for calibration – an unknown phase β and time lag Δt (which also differ across subsequent packets) can distort the CFR. The sanitization module in PinLoc aims to correct for these offsets. The problem arises because the transmitter and the receiver do not attempt to precisely synchronize their timing and their phases (beyond symbol level) before transmitting a packet[3]. Hence, the phase of the channel response of subcarrier f will be $\hat{\phi}_f = \phi_f + 2\pi f_f \Delta t + \beta + Z_f$, where ϕ_f is the genuine channel response phase we are searching for and Z_f is some measurement noise. It is important to notice that we do not need to learn the exact values of Δt and β for each packet (which is probably impossible). Since we feed $\hat{\phi}_f$ to a classification algorithm, we need to make sure that whenever we measure $\hat{\phi}_f$, the measurement includes *the same* values of Δt and β.

We use a simple transform to achieve the goal. For every received channel response we calculate

$$a = \frac{\hat{\phi}_F - \hat{\phi}_1}{2\pi F},$$
$$b = \frac{1}{F} \sum_{1 \le f \le F} \hat{\phi}_f.$$

Intuitively, a is the slope of the received response's phase and b is the offset. It is then easy to verify that, if the measurement noise Z_f is small, $\hat{\phi}_f - af - b$ eliminates the random Δt and β time lags. We verify that this is indeed the case in our measured datasets and we use the post-processed phase value $\hat{\phi}_f - af - b$ in all our further calculations.

---

[3]This is not a problem for a conventional OFDM receiver that only needs to remove, but not learn the channel response.

4.2 Modeling the channel response

We see from a sample measurement, presented in section 3, that the channel responses look like noisy replicas of a few representative clusters. However, it may not be obvious how to identify clusters and the main challenge of the classification algorithm is how to deal with the measurement noise. We make a reasonable assumption to model the noise (also called fast-fading) as a complex Gaussian noise, which corresponds to Rayleigh fading [15]. We first verify this assumption visually by looking at the samples across subcarriers, such as the one illustrated in Figure 2 (b). We then take a few samples from the measurements and verify using QQ plots that the distribution fits well to a complex Gaussian and that the real and the imaginary parts are i.i.d. We also assume that the noise is independent across subcarriers.

Let us consider a link from a single location to a single AP. Recall that $\mathcal{U} = \{\mathbf{U}^1, \cdots, \mathbf{U}^u\}$ is the set of means of the representative CFR clusters of the link, as discussed in section 3, where each $\mathbf{U}^i = (U_f^i)_{f=1,\cdots,F}$. Let us suppose we observe packet $\mathbf{P} = (P_f)_{f=1,\cdots,F}$, where P_f is the complex channel response for subcarrier f.

Following the observations from the measurements, we propose to model the channel response as a random vector with a Gaussian mixture distribution. That is, the channel response is assumed to be drawn from one of the representative CFR clusters, chosen at random for each packet. The channel response is modeled as a combination of the representative CFR clusters. We model each representative CFR cluster as a complex Gaussian random vector with mean $\mathbf{U}^i$ and some variance $\mathbf{V}_i$ (since the real and imaginary parts are i.i.d, the variance is scalar). Assuming that the packet $\mathbf{P}$ belongs to the CFR cluster with the mean $\mathbf{U}^i$, we have that the probability of packet $\mathbf{P}$ is

$$\mathrm{P}(\mathbf{P}|\mathbf{U}^i, \mathbf{V}^i) = \prod_{f=1}^{F} \frac{1}{2\pi \left(V_f^i\right)^2} \exp\left(-\frac{||P_f - U_f^i||^2}{2\left(V_f^i\right)^2}\right). \quad (3)$$

We see that each representative CFR cluster is described with a pair of complex vectors $(\mathbf{U}^i, \mathbf{V}^i)$ representing the mean and the variance of the observations. In section 4.3 we discuss how to derive $(\mathbf{U}^i, \mathbf{V}^i)$ from the measured data set.

Furthermore, we can apply logarithm to (3) and remove constants to derive the log likelihood distance metric

$$d(\mathbf{P}, \mathbf{U}^i) = \sum_{f=1}^{F} \log(V_f^i) + \sum_{f=1}^{F} \left(\frac{||P_f - U_f^i||^2}{\left(V_f^i\right)^2}\right), \quad (4)$$

which we shall use as a distance metric in the classification algorithm. Note that d is indeed a distance metric, but it also has a probabilistic interpretation from (3), which we will use later while combining multiple readings to improve classification accuracy.

4.3 Clustering algorithm

Recall that we model the data at each location as a Gaussian mixture distribution, with K clusters with means and variances (U_k, V_k). We denote with w_k the probability that an observed packet belongs to a particular cluster k, which

corresponds to how often we "see" cluster k in our training data. According to the Gaussian mixture model, this probability is independent across packets. Thus, the parameters of our model are $(w_k, U_k, V_k), k = 1, ..., K$. The classical approach to estimate these parameters is the expectation-maximization algorithm [16]. Instead, we estimate the parameters using variational Bayesian inference [16]. Variational inference is provided by the Infer.NET [17] framework that we use to implement the clustering algorithm. It is particularly convenient here because it tends to prune unneeded clusters. Instead of estimating the number of clusters K by running the algorithm multiple times with different values of K, we perform one run with $K = 10$ and drop the clusters with small weights w_k. Some locations may actually have more than 10 clusters but this is rare and discarding these extra clusters has little impact on performance.

We note that another potential clustering algorithm that could be used here is k-means clustering algorithm. However, our approach takes into account that the noise has a Gaussian distribution and hence can perform a more accurate clustering. For further discussion on the drawbacks of the k-means clustering, please see [16, Chapter 9].

4.4 Classification algorithm

Our classification algorithm is composed of two parts. First, PinLoc computes a macro-location based on WiFi SSIDs alone [8], and shortlists the spots within this macro-location; we call these spots the *candidate set*, C. The second task is to pick one spot from C, or to declare that the device is not in any of these spots. To this end, the WiFi device overhears beacons from the APs as it roams around (we discuss the energy implications in section 5.5). Let A be the set of all APs and let $AP(\mathbf{P})$ denote the AP which transmitted packet $\mathbf{P}$. We define the distance between a given packet $\mathbf{P}$ and a spot S_i as

$$d(\mathbf{P}, S_i) = \min_{\mathbf{U}^i \in Z_i, AP(\mathbf{U}^i) = AP(\mathbf{P})} d(\mathbf{P}, \mathbf{U}^i). \quad (5)$$

where Z_i is a set of representative CFR clusters learned from spot S_i. Then, for all values of i, we compute the minimum of $d(\mathbf{P}, S_i)$ – this outputs the most likely spot that the user is located at, based on the CFRs from packet $\mathbf{P}$. The operation repeats for every packet received within a short time window (typically 30 packets from 3 APs), and the spot that is picked most often (highest vote) is identified. PinLoc does not immediately declare the highest voted spot as the user's location. If the highest vote count is small, it suggests low confidence and the possibility that the user is not located at any of the trained spots[4]. Thus, PinLoc ensures that the number of votes is above a rejection threshold before announcing that spot as the user's location. If the number of votes is below the threshold, then PinLoc announces the location as "*not-a-spot*". The rejection threshold can be selected from the training data and a application-specified false positive rate. We use 15% of the number of possible votes as the threshold in our evaluation. E.g., if the maximum number of votes are 30, PinLoc announces a location as "*not-a-spot*", if the highest matching spot obtain less than 5 votes.

---

[4]We note that, in order to deal with the outliers, we use the majority voting scheme with the distance function (5) instead of the direct probabilistic interpretation (4).

5. EVALUATION

We evaluate PinLoc across 100 different spots using war driven training data and several test samples. Our wardriving approach is explained next.

5.1 War-Driving

The channel model and clustering algorithm from section 4.2 can be applied only to data from the same location. As we have seen in section 3, the size of a location that has the same representative CFR clusters is about 2cm × 2cm. It is important to get enough data from the same location to be able to accurately learn the channel responses. Also, it is important to collect responses from many (2cm × 2cm) locations within the 1m × 1m spot.

Our current war-driving procedure is in 2D. We transmit a packet from an AP every $1ms$. A Roomba robot, mounted with a laptop, moves at a programmed speed of $30cm/s$. Figure 11 shows an example of war-driving at the museum. We receive about 60 packets during a $2cm$ stretch. We divide all packets received in one spot into batches of 60 packets and run clustering algorithm on each batch.

Figure 11: PinLoc war-driving at different spots in the museum. The Roomba robot mounted with a laptop, and 4 virtual wall devices at the corners of the spot.

There are three important questions that arise about war-driving. The first question is *do we need to record every single representative CFR cluster during war-driving?* As we show later in section 5, PinLoc's performance is not too sensitive to war-driving accuracy – the accuracy degrades gracefully as the Roomba is made to war-drive for shorter durations.

The second question is *do we need to war-drive in a particular fashion?* During the localization we match each sample against all learned representative CFR clusters to find the best fit, thus it is not important in which order we store the clusters in the learning database. One can swipe through a spot in any direction, multiple times and in multiple rounds. Moreover, war-driving does not have to be done in any particular channel conditions (e.g. during busy or off-peak hours). In fact, the accuracy increases if we accumulate channel samples from diverse channel conditions.

The third question is *why not define a spot as a single location (2cm × 2cm)?* Recall from Figure 10 that a string of CFRs improves the probability of spot localization. This is because a single test CFR may match with a different random spot; but it is unlikely that a string of test CFRs will all match better with the same wrong spot. In leveraging this diversity gain, we find that strings of length 5 are reasonable. Given that AP beacons are spaced $100ms$ apart in WiFi b/g/n, and human indoor walking speed is $1m/sec$, getting 10 beacons implies that the person moves almost $1m$. Hence, we fix our spot-size to $1m × 1m$. Of course, even if a person is not walking, we assume that there will be inherent swaying motions of the body in the granularity of few cms – such motions will be sufficient for PinLoc.

5.2 Experiment Design

We evaluate PinLoc in 4 environments: (1) Hudson Engineering building with faculty offices and classrooms, (2) busier Bryan Student Center, (3) Twinnies cafe, and (4) Duke University Nasher Museum. Both training and test samples are taken from the laptop, mounted on the Roomba robot. In all the experiments, the laptop associates to all APs on the same frequency channel, and receives beacon packets from them for a duration of $1s$. This $1s$ duration ensures that a mobile user (walking at 1m/s) will remain inside the same spot while receiving all the beacons. For the engineering building scenario, we take test samples during daytime with more than 100 people around. The cafeteria experiments are performed during busy lunch hours, with more than 50 people present at any time, along with a high churn. For the museum measurements PinLoc was trained for 10 large paintings in one wing of the museum, and tested for these spots. The measurements were obtained on a slightly less busy day to minimize interference with real visitors. Both training and test samples are taken at the same height (we discuss the ramifications of height in section 6).

Metrics: Our goal is to show that we can accurately localize test samples to the corresponding spots, but also to detect when a sample does not belong to any spot. We use the following two metrics to evaluate PinLoc. (1) *Accuracy* – the fraction of cases in which the user was localized to the correct spot. (2) *False positives (FP)* – the fraction of cases in which the users were localized to an incorrect spot/non-spot. In other words, false positives also take into account the cases where PinLoc localizes the user to a trained spot, even though the user was not located at any of these spots.

Comparison with RSSI: We also evaluate whether we can use RSSI to achieve a similar accuracy with the same number of APs. For this, we compare PinLoc with a modified Horus algorithm. The original Horus algorithm [4] interpolates RSSI measurements to simplify war-driving. In order to provide a fair comparison, we modify Horus to use the same war-driven training set we use in PinLoc. We define the similarity analogously to (5), replacing correlation with a difference between two RSSI. This is consistent with equation (3) in Horus [4], where the localization metric is a joint probability of seeing different access points. We compare the modified Horus algorithm with PinLoc by using the same test data across the same number of APs.
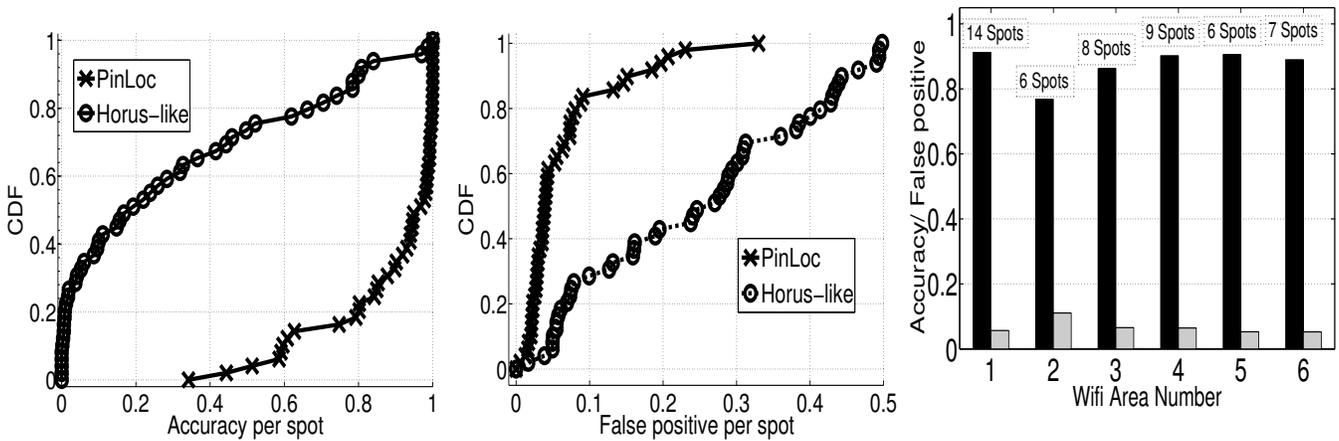
Figure 12: PinLoc in office environment: (a) Accuracy, (b) False positive against Horus (c) Per macro-location.

5.3 PinLoc accuracy and false positives

Figure 12 reports results from the engineering building experiments. PinLoc achieves nearly 90% mean accuracy across 50 spots (Figure 12(a)), consistently outperforming Horus. The false positives (FP) are also maintained to less than 7%, compared to more than 25% in Horus (Figure 12(b)). RSSI based algorithm is significantly worse than PinLoc, since it is represented with a single real number. CFR is represented with 30 complex numbers and contains much richer information. In this comparison, we were tuned to a single channel, and observed 1 to 4 APs on that channel. Ofcourse, PinLoc's, as well as Horus's, performance will improve if more APs are available. But this may require scanning across channels which may increase energy consumption.

Figure 12(c) zooms into the performance of PinLoc and shows the accuracy/FP on a per-WiFi macro location[5]. The number of spots per WiFi region is shown on top of the bars. Student center has a more dynamic environment with cafes and shops – we evaluate PinLoc at 34 spots across 3 WiFi macro locations. Figure 13(a) shows that even in such an active environment PinLoc can maintain low false positive(7.3%) and high accuracy(86%). An obvious question is whether such performance will degrade if adjacent spots need to be identified (i.e., center of spots are $1m$ apart). Figure 13(b) shows that PinLoc is able to discriminate in such settings with a reasonable average accuracy of 82%.

Similar accuracy/FP graphs are plotted for the cafeteria and museum in Figure 14. The mean accuracy for the cafeteria case is 90.07% and the mean FP is 4.5%. For the museum, the mean accuracy is 90.28%, and the mean FP, 4.1%. In all four scenarios, PinLoc achieves high accuracy/low FP for most of the spots, except around 20% where the performance drops. Careful investigation showed that these spots received packets at low SNR from many APs. To probe this further and understand their ramifications, we next perform an analysis across various system parameters.

5.4 Impact of Parameters

---

[5]Recall that a macro-location is derived from WiFi SSIDs; PinLoc discriminates only between candidate spots inside the macro-location.
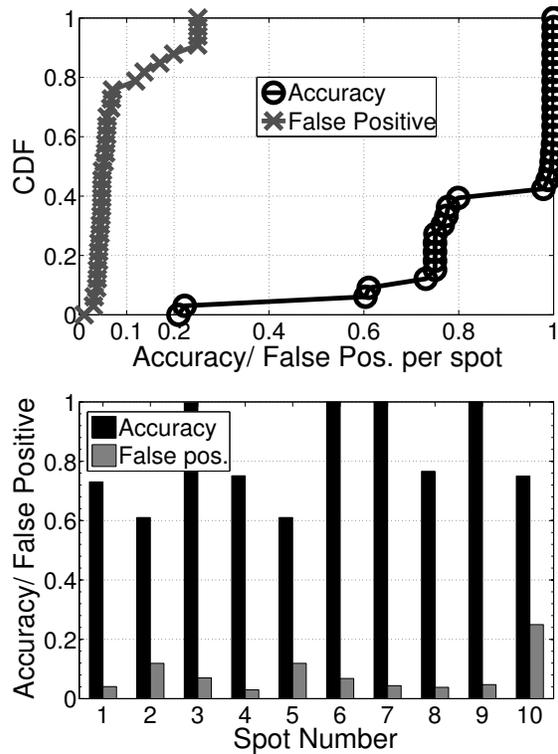
Figure 13: Pinloc performance in student center (a) Accuracy, false pos., (b) Performance of adjacent spots.

Impact of number of test packets

A user extracts CFRs from beacon packets that are transmitted by APs every $100ms$. Thus, assuming up to $1m/s$ walking speeds, the user dwells for at least $1s$ inside a spot, thereby receiving 10 beacons per-AP. Figure 15 (a) shows the variation of accuracy and FP with the number of received beacons per-AP. With the typical size of 10 packets per AP, PinLoc achieves mean accuracy of 89% and false positives of 7% across 50 spots in the engineering building; 15 packets raise them to 91% and 5%, respectively. Figure 15 (a) also shows low accuracy (68%) and high FP (14%) if only 1 packet per AP is used. This is because one single reading may randomly match with an incorrect spot. However, even with 5 packets, we find the accuracy above 82%, and FP less than 10%. This indicates
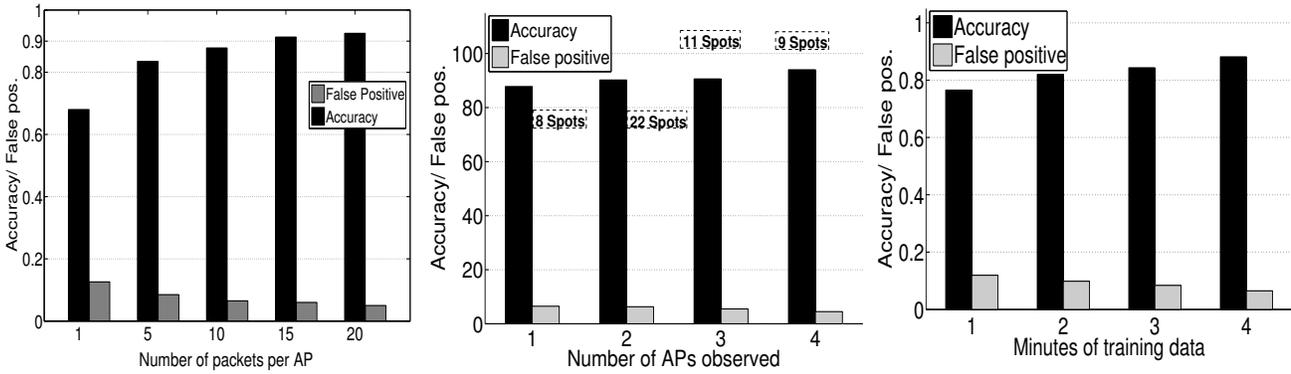
Figure 15: Accuracy of PinLoc against: (a) number of beacons, (b) number of APs and (c) duration of war-driving.
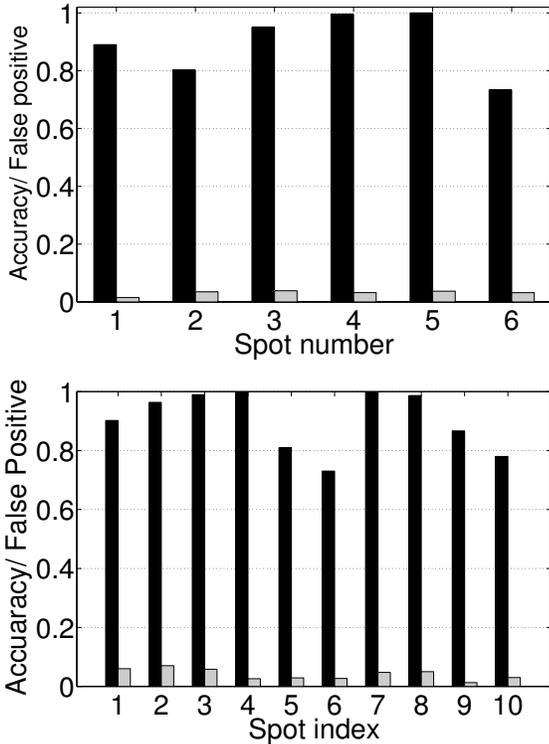
Figure 14: PinLoc performance in cafeteria and museum (a) Accuracy and FP per spot in cafeteria. (b) Accuracy and FP per-spot in the museum.

that even at higher user mobility, or with failure in beacon reception, PinLoc can sustain a reasonably good performance.

Impact of the number of APs

PinLoc's performance in sparse WiFi environments is of interest. To this end, we analyze the performance for varying number of APs. While collecting the test data, we were tuned to a single channel, and observed 1 to 4 APs on that channel (we did not scan across channels to limit energy consumption). We divide the spots into different categories depending on how many APs are visible within each spot – Figure 15 (b) shows the results. An encouraging observation is that *even when only a single AP is visible, PinLoc can perform spot localization with accuracy of over 85% and false positives below 7%.* This is in contrast to other WiFi-based localization meth-

ods that need at least 3 APs to attain reasonable precision. Furthermore, as the number of visible APs increases, the performance improves quickly.

Impact of war-driving

It is important to understand how long we need to war-drive to achieve high localization accuracy. The tradeoff is that short war-driving will record fewer CFRs, incurring the possibility of overlooking an important CFR cluster. To understand the impact, we run PinLoc on different training sets, drawn from different war-driving durations. Figure 15 (c) plots the accuracy and the FP (per-spot) as a function of this duration. Evidently, a few minutes of war-driving per-spot suffices; we observe reasonable performance when using only 1 minute of war-driving data.

Impact of mobility

We turn to the cafeteria scenario to analyze the effect of the mobility on the accuracy of localization. We take one hour of test data for three spots in the cafeteria. We perform localization on each batch of 10 beacon packets and plot its success or failure in Figure 16. For all three spots, we see that the time instants when localization failed are short and uniformly spread over the measurement interval. The mean accuracy was 85% with 7% false positives. Thus, even in a very busy environment such as the cafeteria, we are able to provide localization without prolonged disruption.
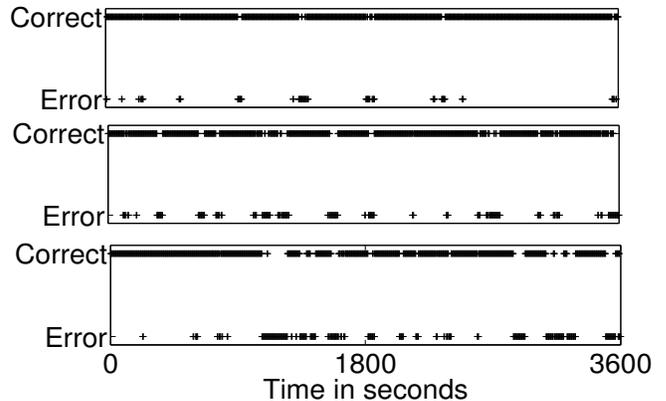
Figure 16: Success of PinLoc localization over time for three spots and over an interval of 1 hour.

Impact of old training data

We concede that PinLoc will need a fresh round of war-driving for spots that are affected by significant environmental changes (e.g., metallic shelves). But with "small" structural changes will PinLoc's war-driven training data remain valid over days and months? To evaluate this we tested 5 spots in our engineering building 7 months after wardriving them. Figure 17 shows a moderate median accuracy of 73% per spot in this scenario. Depending on application requirements, war-driving can be periodically scheduled to improve accuracy. This may not be hard since war-driving can be automatically be done using a Roomba robot.
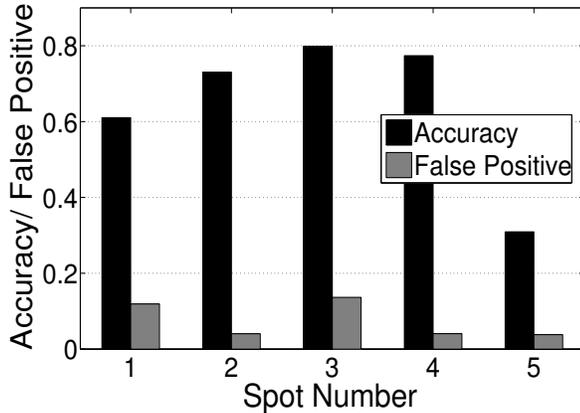
Figure 17: Accuracy of 5 spots tested 7 months after training.

5.5 Energy consumption

PinLoc is designed with energy efficiency in mind. Contrary to existing schemes that rely on power-hungry active scanning, PinLoc uses only beacons from APs in a single channel. For this, it synchronizes with the beacon-schedules of these APs and periodically wakes up to collect the CFRs. It sends this information to a central server for computing location. The amount of such data is low, approximately 1200 bytes per second for 2 APs and $1m/s$ speed, and can be easily batched with existing traffic that communicates to the location based service. Consequently, the data upload energy is marginal. Now, for the energy footprint of beacon reception, we performed measurements on Google Nexus One phones, using the Monsoon Power Meter. We found that receiving beacons from 2 APs every $100ms$ incurs an additional $5.28mW$ power on average. This may be negligible compared to $1326.72mW$ on average to stream YouTube video. We omit the details in the interest of space but argue that PinLoc's low energy overhead makes it a practical proposition.

6. LIMITATIONS AND DISCUSSION

● **Antenna's orientation:** PinLoc's war-driving and testing were all performed with a laptop on a Roomba robot, in a 2D plane. The antenna is placed in the laptop's lid and the lid was closed, hence the antenna was parallel to the testing plane. During wardriving and testing, Roomba robot took random turns, constantly pointing the antenna in different directions. Hence we can conclude our results are robust to the antenna's orientation on the 2D plane.
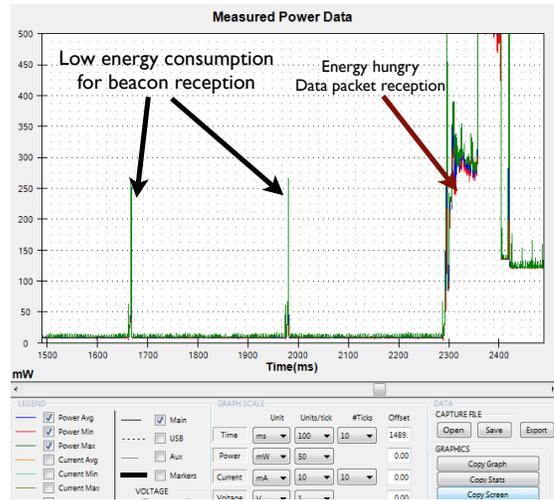
Figure 18: Beacon energy consumption from the Monsoon Power Meter tool.

● **Height, 3D war-driving, and phone mobility:** In reality, users will carry their phones at different heights and 3D war-driving would be necessary. We discuss the effects of 3D wardriving with help of two simple micro-benchmarks.

(1) We manually war-drive a $1m \times 1m \times 50cm$ box for 15 minutes (a person holds a laptop in hand and moves it in random directions across the box), and we collect CFR samples from 4 APs. We then collect test data within the same box by moving the laptop around it and rotating the laptop. The person that holds the laptop also moves around it. We compare the test data with the manually war-driven 3D data and with the previously war-driven 2D data for 9 different locations. We plot the localization accuracy in time in Figure 19(a). The mean accuracy is 84%. We see that the accuracy is similar to the one reported in section 5 hence we can speculate that with suitable extensions, PinLoc might scale real-world scenarios.

(2) To understand the complexity of 3D wardriving, we measure the vertical "size" of a location (the z axis analogue of Figure 8). We see that cross-correlation between CFRs drifts apart at $10cm$ or more (Figure 19(b)). This suggests that 3D war-driving may be feasible, perhaps with a height-varying tripod on top of a Roomba. We were unable to construct a robot for 3D wardriving. We leave 3D localization to future work, as well as issues that may emerge from phone orientation, or users inserting their phones in pockets and bags.

● **Dependency on particular hardware cards:** We run the experiments reported in section 5 alternating among 4 laptops (each with an Intel 5300 card). Since Intel 5300 was the only available card that exposes CFR information, we were not able to test PinLoc with cards from different vendors. We speculate that cross-platform calibration would be necessary, similar to existing RSSI based schemes [7, 18].

● **Localization and MIMO:** We did not explore MIMO capabilities in our current test-bed. A MIMO receiver would provide as many CFR samples as the number of receive antennas, and could be highly valuable for localization. We leave this for future investigation.
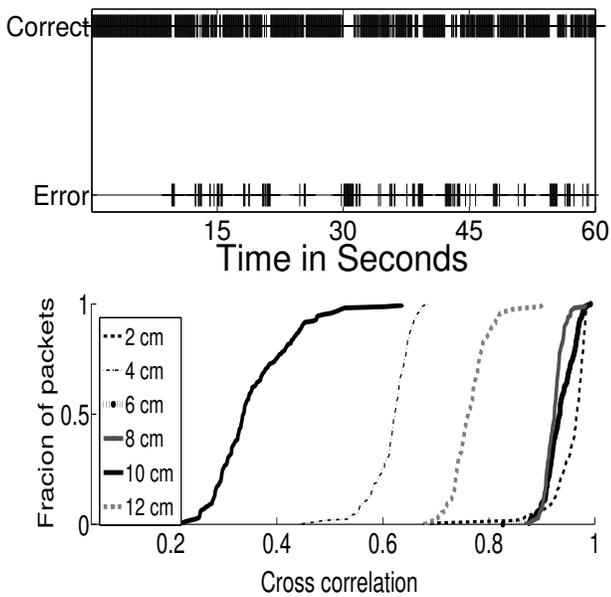
Figure 19: (a) Accuracy of a 3D spot over time. (b) Correlation drifts away with height differences of $10cm$ or more.

7. RELATED WORK

The topic of indoor localization has seen a variety of approaches that may be broadly classified as active and passive, and subclassified into RF and sensing based techniques. We sample some of the key ideas, and discriminate PinLoc from them.

RF signal strength-based localization has been the dominant theme of localization for more than a decade. RADAR [8] performs detailed site surveys a priori and then utilizes the SSIDs and received signal strength reported by wireless devices to generate location fingerprints. Horus [4] and LEASE [19] propose enhancements to RADAR by exploiting the structure of RSSI. Place Lab [6] and the Active Campus project [20] attempt to reduce the overhead of calibration – they show that collecting signal information from WiFi and GSM base stations through war-driving is adequate for reasonably accurate localization. Finally, Patwari and Kasera [11] have also recently explored the use of signal characteristics of a wireless channel to achieve location distinction – that is, they reliably identify when a device has moved from one location to another.

Time-based techniques utilize time delays in signal propagation to estimate distances between wireless transmit-receiver pairs. Examples include GPS [21], PinPoint [22], and work by Werb et. al. [23]. The TPS system uses difference in time of arrival of multiple RF signals from transmitters at known location [24]. Similarly, the PAL system [25, 26] uses time difference of arrival between UWB signals at multiple receivers to determine location. The Cricket system [10, 27] and AHLoS [28] utilize propagation delays between ultrasound and RF signals to estimate location of wireless devices. Such a solution requires installation of ultrasound detectors on wireless devices, limiting their applicability.

Angle-of-arrival based techniques utilize multiple antennas to estimate the angle at which signals are received, and then employ geometric or signal phase relationships to calculate bearings to other devices with respect to the device's own axis [29–31]. Besides positioning, such methods can also provide orientation capabilities. However, these techniques require extremely sophisticated antenna systems (4 to 8 antennas) and non-trivial signal processing capabilities, unforeseeable on mobile devices in the near future. PinLoc's reliance on WiFi alone, along with the ability to utilize PHY layer information from off-the-shelf interfaces [32], makes it a potential candidate for immediate deployment.

8. CONCLUSIONS

This paper shows that PHY layer information, exported by off-the-shelf Intel 5300 cards, offer new opportunities to localize WiFi devices in indoor environments. We leverage the observation that multipath signals exhibit stable patterns in the manner in which they combine at a given location, and these patterns can lend themselves to meter-scale localization. Evaluation results from the engineering building, cafeteria, student center, and the university museum, demonstrate a mean accuracy of 89% for 100 spots. From the application's perspective, PinLoc could enable product advertisements on a shopping aisle, or offer information on each exhibit at a museum. We believe this is a step forward in the area of indoor localization, even though some more work is necessary before it is ready for real-world deployment.

9. ACKNOWLEDGMENT

10. REFERENCES

[1] Placecast. Shopalerts.
    http://placecast.net/shopalerts/index.html.
[2] E. Bruns et al. Enabling mobile phones to support large-scale museum guidance. *Multimedia, IEEE*, 2007.
[3] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *MOBICOM*, 2000.
[4] M. Youssef and A. Agrawala. The horus WLAN location determination system. In *MobiSys*, 2005.
[5] Chen Y. et al. Fm-based indoor localization. In *Mobisys*. ACM, 2012.
[6] Yu-Chung Cheng, Yatin Chawathe, Anthony LaMarca, and John Krumm. Accuracy characterization for metropolitan-scale wi-fi localization. In *MobiSys*, 2005.
[7] K. Chintalapudi, A. Iyer, and V. Padmanabhan. Indoor localization without the pain. In *MOBICOM*, 2010.
[8] V. Bahl et al. RADAR: An in-building rf-based user location and tracking system. In *INFOCOM*, 2000.
[9] I. Constandache, S. Gaonkar, M. Sayler, R. Roy Choudhury, and L. Cox. Enloc: Energy-efficient localization for mobile phones. In *IEEE Infocom Mini Conference*, 2009.
[10] N. B. Priyantha. *The cricket indoor location system*. PhD thesis, MIT, 2005.
[11] Junxing Zhang et al. Advancing wireless link signatures for location distinction. In *MobiCom*. ACM, 2008.
[12] M. Azizyan, I. Constandache, and R. R. Choudhury.

SurroundSense: Mobile phone localization via ambience fingerprinting. In *MOBICOM*, 2009.

[13] A. Sheth, S. Seshan, and D. Wetherall. Geo-fencing: Confining Wi-Fi coverage to physical boundaries. *Pervasive Computing*, 2009.

[14] Intel Research. Intel 5300 mimo channel measurement tool. `http://ils.intel-research.net/80211n-channel-measurement-tool`.

[15] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[16] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[17] T. Minka, J.M. Winn, J.P. Guiver, and D.A. Knowles. Infer.NET 2.4, 2010. Microsoft Research Cambridge. http://research.microsoft.com/infernet.

[18] A. Haeberlen et al. Practical robust localization over large-scale 802.11 wireless networks. In *MOBICOM*. ACM, 2004.

[19] P. Krishnan and others. A system for LEASE: Location estimation assisted by stationary emitters for indoor rf wireless networks. In *IEEE Infocom*, March 2004.

[20] William G. Griswold et al. Activecampus: Experiments in community-oriented ubiquitous computing. *Computer*, 2004.

[21] The global positioning systems (wikipedia entry). http://en.wikipedia.org/wiki/Global_Positioning_System.

[22] M. Youssef et al. Pinpoint: An asynchronous time-based location determination system. In *ACM Mobisys*, June 2006.

[23] J. Werb and C. Lanzl. Designing a positioning system for finding things and people indoors. *IEEE Spectrum*, 35(9), September 1998.

[24] X. Cheng et al. TPS: A time-based positioning scheme for outdoor sensor networks. In *IEEE Infocom*, March 2004.

[25] S.J. Fontana, R.J.and Gunderson. Ultra-wideband precision asset location system. In *IEEE Conference on Ultra Wideband Systems and Technologies*, May 2002.

[26] R.J. Fontana, E. Richley, and J. Barney. Commercialization of an ultra wideband precision asset location system. In *IEEE Conference on Ultra Wideband Systems and Technologies*, November 2003.

[27] A. Smith et al. Tracking moving devices with the cricket location system. In *ACM Mobisys*, June 2004.

[28] A. Savvides and M. Han, C.C. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *ACM MobiCom*, 2001.

[29] D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AoA. In *IEEE Infocom*, 2003.

[30] D. Niculescu and B. Nath. VOR base stations for indoor 802.11 positioning. In *ACM MobiCom*, September 2004.

[31] J. Xiong and K. Jamieson. SecureAngle: improving wireless security using angle-of-arrival information. In *ACM HotNets*, 2010.

[32] S. Sen et al. Precise Indoor Localization using PHY Layer Information. In *ACM HotNets*, 2011.