ArrayTrack: A Fine-Grained Indoor Location System

Jie Xiong and Kyle Jamieson
University College London

Abstract

With myriad augmented reality, social networking, and retail shopping applications all on the horizon for the mobile handheld, a fast and accurate location technology will become key to a rich user experience. When roaming outdoors, users can usually count on a clear GPS signal for accurate location, but indoors, GPS often fades, and so up until recently, mobiles have had to rely mainly on rather coarse-grained signal strength readings. What has changed this status quo is the recent trend of dramatically increasing numbers of antennas at the indoor access point, mainly to bolster capacity and coverage with multiple-input, multiple-output (MIMO) techniques. We thus observe an opportunity to revisit the important problem of localization with a fresh perspective. This paper presents the design and experimental evaluation of ArrayTrack, an indoor location system that uses MIMO-based techniques to track wireless clients at a very fine granularity in real time, as they roam about a building. With a combination of FPGA and general purpose computing, we have built a prototype of the ArrayTrack system. Our results show that the techniques we propose can pinpoint 41 clients spread out over an indoor office environment to within 23 centimeters median accuracy, with the system incurring just 100 milliseconds latency, making for the first time ubiquitous real-time, fine-grained location available on the mobile handset.

1 Introduction

The proliferation of mobile computing devices continues, with handheld smartphones, tablets, and laptops a part of our everyday lives. Outdoors, these devices largely enjoy a robust and relatively accurate location service from Global Positioning System (GPS) satellite signals, but indoors where GPS signals don't reach, providing an accurate location service is quite challenging.

Furthermore, the demand for accurate location information is especially acute indoors. While the few meters of accuracy GPS provides outdoors are more than sufficient for street-level navigation, small differences in location have more importance to people and applications indoors: a few meters of error in estimated location can place someone in a different office within a building, for example. Location-aware smartphone applications on the horizon such as augmented reality-based building navigation, social networking, and retail shopping demand both a high accuracy and a low response time. A solution that offers a centimeter-accurate location service indoors would enable these and other exciting applications in mobile and pervasive computing.

Using radio frequency (RF) for location has many challenges. First, the many objects found indoors near access points (APs) and mobile clients reflect the energy of the wireless signal in a phenomenon called *multipath propagation*. This forces an unfortunate tradeoff that most existing RF location-based systems make: either model this hard-to-predict pattern of multipath fading, or leverage expensive hardware that can sample the wireless signal at a very high rate. Most existing RF systems choose the former, building maps of multipath signal strength [2, 3, 34, 43], or estimating coarse differences using RF propagation models [11, 14], achieving an average localization accuracy of between 60 cm [43] and meters: too coarse for the applications at hand.

Systems based on ultrasound and RF sensors such as Active Badge [35], Bat [36], and Cricket [19] have achieved accuracy to the level of centimeters, but usually require dedicated infrastructure to be installed in every room in a building, an approach that is expensive, time consuming, and requires maintenance effort.

Recently, however, two new opportunities have arisen in the design of indoor location systems:

1. WiFi APs are incorporating ever-increasing numbers of antennas to bolster capacity and coverage with multiple-input, multiple-output (MIMO) techniques. In fact, we expect that in the future, the number of antennas at the access point will increase several-fold, to meet the demand for MIMO links and spatial multiplexing [1, 31], which increase overall capacity. Indeed, the upcoming 802.11ac standard will specify eight MIMO spatial streams, and 16-antenna APs have been available since 2010 [41].

2. Meanwhile, WiFi AP density remains high: With our experimental infrastructure excluded, transmissions from most locations in our testbed reach seven or more production network APs, with all but about five percent of locations reaching five or more such APs. Furthermore, by leveraging the signal processing that is possible at the physical layer, an AP can extract information from a single packet at a lower SNR than

This material is based on work supported by the European Research Council under Grant No. 279976. Jie Xiong is supported by the Google European Doctoral Fellowship in Wireless Networking.

what is required to receive and decode the packet. This allows even more ArrayTrack APs to cooperate to localize clients than would be possible were the system to operate exclusively at higher layers.

ArrayTrack is an indoor localization system that exploits the increasing number of antennas at commodity APs to provide fine-grained location for mobile devices in an indoor setting. When a client transmits a frame on the air, multiple ArrayTrack APs overhear the transmission, and each compute angle-of-arrival (AoA) information from the clients' incoming frame. Then, the system aggregates the APs' AoA data at a central backend server to estimate the client's location. While AoA techniques are already in wide use in radar and acoustics, the challenge in realizing these techniques indoors is the presence of strong multipath RF propagation in these environments. To address this problem, we introduce novel system design techniques and signal processing algorithms that reliably eliminate the effects of multipath, even in the relatively common situations when little or no energy arrives on the direct path between client and AP.

ArrayTrack advances the state of the art in localization in three distinct ways:

1. To mitigate the effects of indoor multipath propagation, ArrayTrack contributes a novel *multipath suppression* algorithm to effectively remove the reflection paths between clients and APs.
2. Upon detecting a frame, an ArrayTrack AP quickly switches between sets of antennas, synthesizing new AoA information from each. We term this technique *diversity synthesis*, and find that it is especially useful in the case of low AP density.
3. ArrayTrack's system architecture centers around parallel processing in hardware, at APs, and in software, at the database backend, for fast location estimates.

We implement ArrayTrack on the Rice WARP FPGA platform and evaluate in a 41-node network deployed over one floor of a busy office space. Experimental results in this setting show that using just three APs, ArrayTrack can localize clients to a median 57 cm and mean one meter accuracy. With six APs, ArrayTrack achieves a median 23 cm and mean 31 cm location accuracy, localizing 95% of clients to within 90 cm. At the same time, ArrayTrack is fast, requiring just 100 milliseconds to produce a location estimate. To our knowledge, these are the most accurate and responsive location estimates available to date for an RF-based location system that does not require infrastructure except a normal density of nearby WiFi APs. Furthermore, as we experimentally show, ArrayTrack's performance is robust against different antenna heights, different antenna orientation, low SNR and collisions.

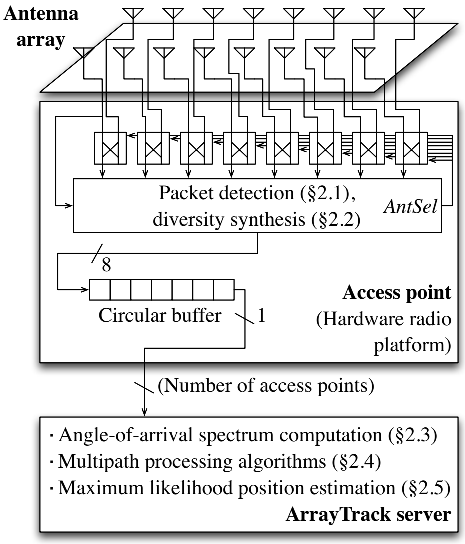

Antenna array
Packet detection (§2.1), diversity synthesis (§2.2)
AntSel
8
Circular buffer
1
Access point
(Hardware radio platform)
(Number of access points)
· Angle-of-arrival spectrum computation (§2.3)
· Multipath processing algorithms (§2.4)
· Maximum likelihood position estimation (§2.5)
ArrayTrack server

Figure 1: ArrayTrack's high-level design for eight radio front-ends, divided into functionality at each ArrayTrack access point and centralized server functionality. For clarity, we omit transmit path functionality of the access point.

In the next section, we detail ArrayTrack's design. An implementation discussion (§3) and performance evaluation (§4) then follow. We discuss related work in Section 5, and Section 7 concludes.

2 Design

We describe ArrayTrack's design as information flows through the system, from the physical antenna array, through the AP hardware, and on to the central ArrayTrack server, as summarized in Figure 1. First, ArrayTrack leverages techniques to detect packets at very low signal strength, so that many access points can overhear a single transmission (§2.1). Next, at each AP, ArrayTrack uses many antennas (§2.2) to generate an *AoA spectrum*: an estimate of likelihood versus bearing (§2.3), and cancels some of the effects of multipath propagation (§2.4). Finally, the system combines these estimates to estimate location (§2.5), further eliminating multipath's effects.

2.1 Packet detection and buffer management

To compute an AoA spectrum for a client, the AP only need overhear a small number of frames (between one and three, for reasons that will become clear in Section 2.4) from that client. For ArrayTrack's purposes, the contents of the frame are immaterial, so our system can process control frames such as acknowledgments, and even frames encrypted at the link layer.

The physical-layer preamble of an 801.11 frame contains known short and long training symbols, as shown in Figure 2. We use a modified version of Schmidl-Cox [25] detection to detect an incoming frame's short training symbols. Once the detection block senses a frame, it activates the diversity synthesis mechanism described in

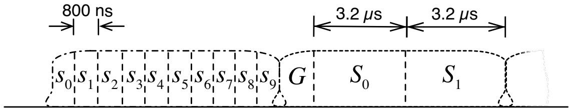


Figure 2: The 802.11 OFDM preamble consists of ten identical, repeated *short training symbols* (denoted $s_0 \ldots s_9$), followed by a *guard interval* (denoted G), ending with two identical, repeated *long training symbols* (denoted S_0 and S_1).

the next section and stores the samples of the incoming frame into a circular buffer, one logical buffer entry per frame detected.

Since it does not require even a partial packet decode, our system can process any part of the packet, which is helpful in the event of collisions in a carrier-sense multiple access network (§4.3.5). Our system detects the preamble of the packet and records a small part of it. In principle, one time domain packet sample would provide enough information for the AoA spectrum computation described in Section 2.3. However, to average out the effects of noise, we use 10 samples (we justify this choice in Section 4.3.3). Since a commodity WiFi AP samples at 40 Msamples/second, this implies that we need to process just 250 nanoseconds of a packet's samples, under 1.5% of an WiFi preamble's 16 μs duration.

2.2 Diversity synthesis

Upon detecting a packet, most commodity APs switch between pairs of antennas selecting the antenna from each pair with the strongest signal, a technique called *diversity selection*. This well-known and widely implemented technique improves performance in the presence of destructive multipath fading at one of the antennas, and can be found in the newest 802.11n MIMO access points today. It also has the advantage of not increasing the number of radios required, thus saving cost at the AP.

ArrayTrack seamlessly incorporates diversity selection into its design, synthesizing independent AoA data from both antennas of the diversity pair. We term this technique *diversity synthesis*.

Referring to Figure 1, once the packet detection block has indicated the start of a packet, control logic stores the samples corresponding to the preamble's long training symbol S_0 (Figure 2) from the upper set of antennas into the first half of a circular buffer entry. Next, the control logic toggles the *AntSel* (for antenna select) line in Figure 1, switching to the lower set of antennas for the duration of the second long training symbol S_1.[1] Since S_1 and S_2 are identical and each 3.2 μs long, they fall well within the coherence time[2] of the indoor wireless channel, and so we can treat the information obtained from each set of antennas as if the two sets were obtained simultaneously from different radios at the AP.

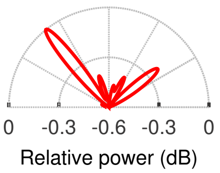


Figure 3: The *AoA spectrum* of a client's received signal at a multi-antenna access point estimates the incoming signal's power as a function of its angle of arrival.

2.3 AoA spectrum generation

Especially in indoor wireless channels, RF signals reflect off objects in the environment, resulting in multiple copies of the signal arriving at the access point: this phenomenon is known as multipath propagation. An AoA spectrum of a client's received signal at a multi-antenna AP is an estimate of the incoming signal's power as a function of angle of arrival, as shown in Figure 3. Since strong multipath propagation is present indoors, the direct-path signal may be significantly weaker than the reflected-path signals, or may even be undetectable. In these situations, the highest peak on the AoA spectrum would correspond to a reflected path instead of the direct path to the client. This makes indoor localization using AoA spectra alone highly inaccurate, necessitating the remaining steps in ArrayTrack's processing chain.

2.3.1 Phased-array primer

In order to explain how we generate AoA spectra, we now present a brief primer on phased arrays. While their technology is well established, we focus on indoor applications, highlighting the resulting complexities.

For clarity of exposition, we first describe how an AP can compute angle of arrival information in free space (*i.e.*, in the absence of multipath reflections), and then generalize the principles to handle multipath wireless propagation. The key to computing a wireless signal's angle of arrival is to analyze received *phase* at the AP, a quantity that progresses linearly from zero to 2π every RF wavelength λ along the path from client to access point, as shown in Figure 4 (*left*).

This means that when the client sends a signal, the AP receives it with a phase determined by the path length d from the client. Phase is particularly easy to measure at the physical layer, because software-defined and hardware radios represent the phase of the wireless signal graphically using an *in-phase-quadrature* (I-Q) plot, as shown in Figure 4 (*right*), where angle measured from the I axis indicates phase. Using the I-Q plot, we see that a distance d adds a phase of $2\pi d/\lambda$ as shown by the angle

[1] We use the long training symbols because our hardware radio platform has a 500 ns switching time during which the received signal is highly distorted and unusable.

[2] The time span over which the channel can be considered stationary. Coherence time is mainly a function of the RF carrier frequency and speed of motion of the transmitter, receiver, and any nearby objects.

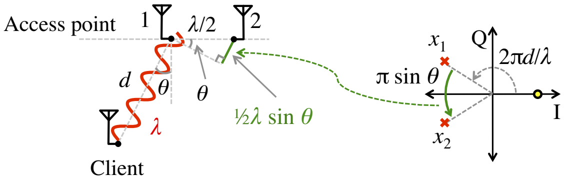


Figure 4: Principle of operation for ArrayTrack's AoA spectrum computation phase. (*Left:*) The phase of the signal goes through a 2π cycle every radio wavelength λ, and the distance differential between the client and successive antennas on the access point is related to the client's bearing on the access point. (*Right:*) The complex representation of the sent signal at the client (filled dot) and received signals at the access point (crosses) reflects this relationship.

measured from the I axis to the cross labeled x_1 (representing the signal received at antenna one). Since there is a $\lambda/2$ separation between the two antennas, the distance along a path arriving at bearing θ is a fraction of a wavelength greater to the second antenna than it is to the first, that fraction depending on θ. Assuming $d \gg \lambda/2$, the added distance is $\frac{1}{2}\lambda \sin\theta$.

These facts suggest a particularly simple way to compute θ at a two-antenna access point in the absence of multipath: measure x_1 and x_2 directly, compute the phase of each ($\angle x_1$ and $\angle x_2$), then solve for θ as

$$\theta = \arcsin\left(\frac{\angle x_2 - \angle x_1}{\pi}\right) \tag{1}$$

Generalizing to multiple antennas. In indoor multipath environments, Equation 1 quickly breaks down, because multiple paths' signals sum in the I-Q plot. However, adding multiple, say M, antennas can help. The best known algorithms are based on eigenstructure analysis of an $M \times M$ *correlation matrix* $\mathbf{R_{xx}}$ in which the entry at the l^{th} column and m^{th} row is the mean correlation between the l^{th} and m^{th} antennas' signals.

Suppose D signals $s_1(t), \dots, s_D(t)$ arrive from bearings $\theta_1, \dots, \theta_D$ at $M > D$ antennas, and that $x_j(t)$ is the received signal at j^{th} antenna element at time t. Recalling the relationship between measured phase differences and AP bearing discussed above, we use the *array steering vector* $\mathbf{a}(\theta)$ to characterize how much added phase (relative to the first antenna) we see at each of the other antennas, as a function of the incoming signal's bearing. For a linear array,

$$\mathbf{a}(\theta) = \exp\left(\frac{-j2\pi d}{\lambda}\right)\begin{bmatrix} 1 \\ \exp(-j\pi\lambda\cos\theta) \\ \exp(-j2\pi\lambda\cos\theta) \\ \vdots \\ \exp(-j(M-1)\pi\lambda\cos\theta) \end{bmatrix} \tag{2}$$

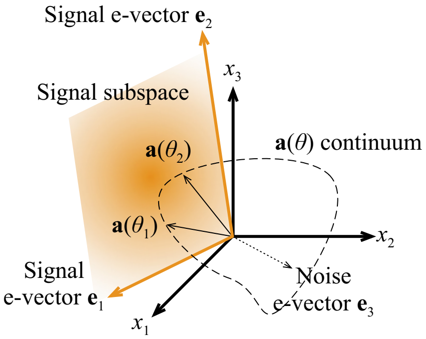


Figure 5: In this three-antenna example, the two incoming signals (at bearings θ_1 and θ_2 respectively) lie in a three-dimensional space. Eigenvector analysis identifies the two-dimensional *signal subspace* shown, and MUSIC traces along the array steering vector continuum measuring the distance to the signal subspace. Figure adapted from Schmidt [26].

So we can express what the AP hears as

$$\mathbf{x}(t) = \overbrace{[\mathbf{a}(\theta_1)\ \mathbf{a}(\theta_2)\ \cdots\ \mathbf{a}(\theta_D)]}^{\mathbf{A}}\begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_D(t) \end{bmatrix} + n(k), \tag{3}$$

where $\mathbf{n}(k)$ is noise with zero mean and σ_n^2 variance. This means that we can express $\mathbf{R_{xx}}$ as

$$\begin{aligned} \mathbf{R_{xx}} &= \mathbb{E}[\mathbf{xx}^*] \\ &= \mathbb{E}\left[(\mathbf{As}+\mathbf{n})\left(\mathbf{s}^*\mathbf{A}^*+\mathbf{n}^*\right)\right] \\ &= \mathbf{A}\mathbb{E}\left[\mathbf{ss}^*\right]\mathbf{A}^* + \mathbb{E}\left[\mathbf{nn}^*\right] \\ &= \mathbf{AR_{ss}A}^* + \sigma_n^2\mathbf{I} \end{aligned} \tag{4}$$

where $\mathbf{R_{ss}} = \mathbb{E}\left[\mathbf{ss}^*\right]$ is the source correlation matrix.

The array correlation matrix $\mathbf{R_{xx}}$ has M eigenvalues $\lambda_1, \dots, \lambda_M$ associated respectively with M eigenvectors $\mathbf{E} = [\mathbf{e}_1\ \mathbf{e}_2\ \cdots\ \mathbf{e}_M]$. If the noise is weaker than the incoming signals, then when the eigenvalues are sorted in non-decreasing order, the smallest $M-D$ correspond to the noise while the next D correspond to the D incoming signals. The D value depends on how many eigenvalues are considered big enough to be signals. We choose D value as how many eigenvalues are larger than a threshold that is a fraction of the largest eigenvalue. Based on this process, the corresponding eigenvectors in $\mathbf{E}$ can be classified as noise or signal:

$$\mathbf{E} = \left[\overbrace{e_1\ \dots\ e_{M-D}}^{\mathbf{E}_N}\ \overbrace{e_{M-D+1}\ \dots\ e_M}^{\mathbf{E}_S}\right] \tag{5}$$

we refer to $\mathbf{E}_N$ as the *noise subspace* and $\mathbf{E}_S$ as the *signal subspace*.

The MUSIC AoA spectrum [26] inverts the distance between a point moving along the array steering vector

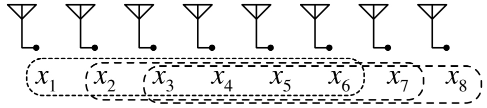


Figure 6: Spatial smoothing an eight-antenna array $x_1, \ldots, x_8$ to a virtual six-element array (number of groups $N_G = 3$).

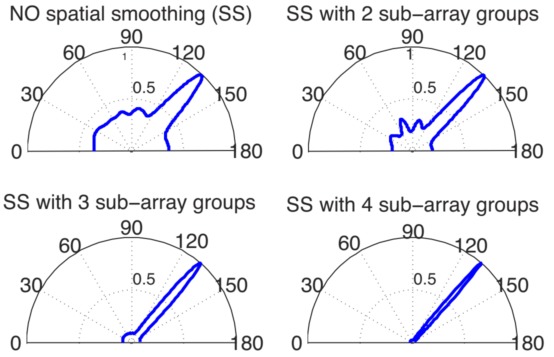


Figure 7: Varying the amount of spatial smoothing on AoA spectra.

continuum and $\mathbf{E}_S$, as shown in Figure 5:

$$P(\theta) = \frac{1}{\mathbf{a}(\theta)\mathbf{E}_N\mathbf{E}_N^*\mathbf{a}(\theta)} \tag{6}$$

This yields sharp peaks in $P(\theta)$ at the signals' AoA.

2.3.2 *Spatial smoothing for multipath distortion*

Implementing MUSIC as-is, however, yields highly distorted AoA spectra, for the following reason. When the incoming signals are phase-synchronized with each other (as results from multipath) MUSIC perceives the distinct incoming signals as one superposed signal, resulting in false peaks in $P(\theta)$. We therefore adopt *spatial smoothing* [28], averaging incoming signals across N_G groups of antennas to reduce this correlation. For example, spatial smoothing over $N_G = 3$ six-antenna groups on an eight-antenna array generating signals $x_1, \ldots, x_8$ would output six signals $\dot{x}_1, \ldots, \dot{x}_6$ where $\dot{x}_i = \frac{1}{3}(x_i + x_{i+1} + x_{i+2})$ for $1 \leq i \leq 6$, as shown in Figure 6.

How should we choose N_G? Figure 7 shows a microbenchmark computing MUSIC AoA spectra for a client near and in the line of sight of the AP (so the direct-path bearing dominates $P(\theta)$) both with and without spatial smoothing. As N_G increases, the effective number of antennas decreases, and so spatial smoothing can eliminate smaller peaks that may correspond to the direct path. On the other hand, as N_G increases, the overall noise in the AoA spectrum decreases, and some peaks may be narrowed, possibly increasing accuracy. Based on this microbenchmark and experience generating AoA spectra indoors from a number of different clients in our testbed, we find that a good compromise is to set $N_G = 2$, and we use this in the performance evaluation in Section 4.

Scenario	**Frequency**
Direct path same; reflection paths changed	71%
Direct path same; reflection paths same	18%
Direct path changed; reflection paths changed	8%
Direct path changed; reflection paths same	3%

Table 1: Peak stability microbenchmark measuring the frequency of the direct and reflection-path peaks changing due to slight movement.

2.3.3 *Array geometry weighting*

Information from the linear array we use in our system is not equally reliable as a function of θ, because of the asymmetric physical geometry of the array. Consequently, after computing a spatially-smoothed MUSIC AoA spectrum, the ArrayTrack multiplies it by a *windowing function* $W(\theta)$, the purpose of which is to weight information from the AoA spectrum in proportion to the confidence that we have in the data. With a linear array, we multiply $P(\theta)$ by

$$W(\theta) = \begin{cases} 1, & \text{if } 15^\circ < |\theta| < 165^\circ \\ \sin\theta, & \text{otherwise.} \end{cases} \tag{7}$$

2.3.4 *Array symmetry removal*

Although a linear antenna array can determine bearing, it cannot determine which side of the array the signal is arriving from. This means that the AoA spectrum is essentially a 180° spectrum mirrored to 360°. When there are many APs cooperating to determine location, this is not a problem, but when there are few APs, accuracy suffers. To address this, we employ the diversity synthesis scheme described in Section 2.2 to have a ninth antenna not in the same row as the other eight included. Using the ninth antenna, we calculate the total power on each side, and remove the half with less power, resulting in a true 360° AoA spectrum.

2.4 Multipath suppression

While the spatial smoothing algorithms described above (§2.3.2) reduce multipath-induced distortion of the AoA spectrum to yield an accurate spectrum, they don't identify the direct path, leaving multipath reflections free to reduce system accuracy. The multipath suppression algorithm we present here has the goal of removing or reducing peaks in the AoA spectrum not associated with the direct path from AP to client.

ArrayTrack's multipath suppression algorithm leverages changes in the wireless channel that occur when the transmitter or objections in the vicinity move by grouping together AoA spectra from multiple frames, if available. Our method is motivated by the following observation: when there are small movements of the transmitter, the receiver, or objects between the two, the direct-path peak on the AoA spectrum is usually stable while the reflection-path peaks usually change significantly, and

Algorithm (Multipath suppression)

1. Group two to three AoA spectra from frames spaced closer than 100 ms in time; if no such grouping exists for a spectrum, then output that spectrum to the synthesis step (§2.5).
2. Arbitrarily choose one AoA spectrum as the *primary*, and remove peaks from the primary not paired with peaks on other AoA spectra.
3. Output the primary to the synthesis step (§2.5).

Figure 8: ArrayTrack's multipath suppression algorithm.

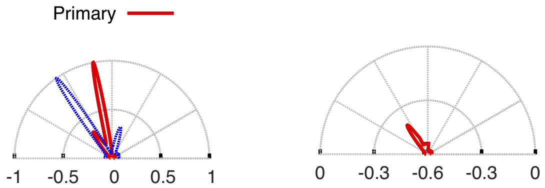


Figure 9: ArrayTrack's multipath suppression algorithm operating on two example AoA spectra (*left*) and the output AoA spectrum (*right*).

these slight movements happen frequently in real life when we hold a mobile handset making calls, for example.

We run a microbenchmark at 100 randomly chosen locations in our testbed (see Figure 12, p. 7), generating AoA spectra at each position selected and another position five centimeters away. If the corresponding bearing peaks of the two spectra are within five degrees, we mark that bearing as *unchanged*. If the variation is more than five degrees or the peak vanishes, we mark it *changed*.

The results are shown in Table 1. Most of the time, the direct-path peak is unchanged while the reflection-path peaks are changed. This motivates the algorithm shown in Figure 8. Note that for those scenarios in which both the direct-path and reflection-path peaks are unchanged, we keep all of them without any deleterious consequences. Also, observe that the microbenchmark above only captures two packets. This leaves room for even further improvement if we capture multiple packets during the course of the mobile's movement. The only scenario which induces failure in the multipath suppression algorithm is when the reflection-path peaks remain unchanged while the direct-path peak is changed. However, as shown above, the chances of this happening are small. We show and example of the algorithm's operation in Figure 9.

2.5 AoA spectra synthesis

In this step, ArrayTrack combines the AoA spectra of several APs into a final location estimate. Suppose N APs generate AoA spectra $P_1(\theta), \ldots, P_N(\theta)$ as processed by the previous steps, and we wish to compute the likelihood of the client being located at position $\mathbf{x}$ as shown in Figure 3. ArrayTrack computes the bearing of $\mathbf{x}$ to AP i, θ_i, by trigonometry, and then estimates the likelihood of the client being at location $\mathbf{x}$, $L(\mathbf{x})$, as

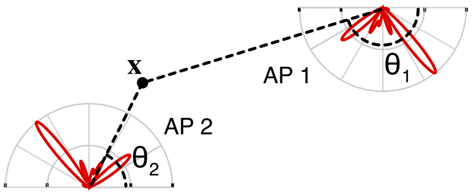


Figure 10: ArrayTrack combines information from multiple APs into a likelihood of the client being at location **x** by considering all AoA spectra at their respective bearings (θ_1, θ_2) to **x**.

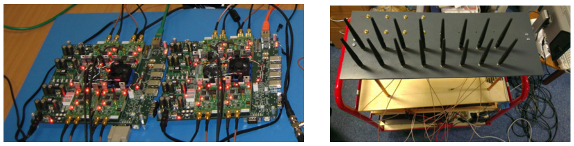


Figure 11: *Left:* the ArrayTrack prototype AP is composed of two WARP radios, while a cable-connected USRP2 software-defined radio (not shown) calibrates the array. *Right:* The AP mounted on a cart, showing its antenna array.

$$L(\mathbf{x}) = \prod_{i=1}^{N} P_i(\theta_i). \tag{8}$$

With Equation 8 we search for the most likely location of the client by forming a 10 centimeter by 10 centimeter grid, and evaluating $L(\mathbf{x})$ at each point in the grid. We then use hill climbing on the three positions with highest $L(\mathbf{x})$ in the grid using the gradient defined by Equation 8 to refine our location estimate.

3 Implementation

The prototype ArrayTrack AP, shown in Figure 11, uses two Rice WARP FPGA-based wireless radios. Each WARP is equipped with four radio front ends and four omnidirectional antennas. We utilize the digital I/O pins on one of the WARP boards to output a time synchronization pulse on a wire connected between the two WARPs, so that the second WARP board can record and buffer the same time-indexed samples as the first. The WARPs run a custom FPGA hardware design architected with Xilinx System Generator for DSP that implements all the functionality described in Section 2.

We place the 16 antennas[3] attached to the WARP radios in a rectangular geometry (Figure 11, right). Antennas are spaced at a half wavelength distance (6.13 cm) to yield maximum AoA spectrum resolution. This also happens to yield maximum MIMO wireless capacity, and so is the arrangement preferred in commodity APs.

AP phase calibration. Equipping the AP with multiple antennas is necessary for ArrayTrack, but does not

[3] The two WAPPs have a total of eight radio boards, each with two ports. ArrayTrack is able to switch ports as described in §2.2 and record the two long training symbols with different antennas. So with two WARPs, the maximum number of antennas we can utilize is 16.

suffice to calculate angle of arrival as described in the preceding section. Each radio receiver incorporates a 2.4 GHz oscillator whose purpose is to convert the incoming radio frequency signal to its representation in I-Q space shown, for example, in Figure 4 (p. 4). An undesirable consequence of this *downconversion* step is that it introduces an unknown phase offset to the resulting signal, rendering AoA inoperable. This is permissible for MIMO, but not for our application, because this manifests as an unknown phase added to the constellation points in Figure 4. Our solution is to calibrate the array with a USRP2 generating a continuous wave tone, measuring each phase offset directly. Because small manufacturing imperfections exist for SMA splitters and cables labelled the same length, we propose a one-time (run only once for a particular set of hardware) calibration scheme to handle these equipment imperfections.

The signal from the USRP2 goes through splitters and cables (we called them external paths) before reaching WARP radios. The phase offset Ph_{off} we want to measure is the internal phase difference $Ph_{in2} - Ph_{in1}$. Running calibration once, we obtain the following offset:

$$Ph_{off1} = (Ph_{ex2} + Ph_{in2}) - (Ph_{ex1} + Ph_{in1}) \qquad (9)$$

Because of equipment imperfections, Ph_{ex2} is slightly different from Ph_{ex1} so Ph_{off1} is not equal to Ph_{off}. We exchange the external paths and run calibration again:

$$Ph_{off2} = (Ph_{ex1} + Ph_{in2}) - (Ph_{ex2} + Ph_{in1}) \qquad (10)$$

Combing the above two equations, we obtain Ph_{off} and the phase difference caused by equipment imperfections:

$$(Ph_{off2} + Ph_{off1})/2 = Ph_{off} \qquad (11)$$

$$(Ph_{off2} - Ph_{off1})/2 = Ph_{ex1} - Ph_{ex2} \qquad (12)$$

Subtracting the measured phase offsets from the incoming signals over the air then cancels the unknown phase difference, and AoA becomes possible.

Testbed clients. The clients we use in our experiments are Soekris boxes equipped with Atheros 802.11g radios operating in the 2.4 GHz band.

4 Evaluation

To show how well ArrayTrack performs in real indoor environment, we present experimental results from the testbed described in Section 3. First we present the accuracy level ArrayTrack achieves in the challenging indoor office environment and explore the effects of number of antennas and number of APs on the performance of ArrayTrack. After that, we demonstrate that ArrayTrack is robust against different transmitter/receiver heights and different antenna orientations between clients and APs. Finally we examine the latency introduced by ArrayTrack, which is a critical factor for a real-time system.

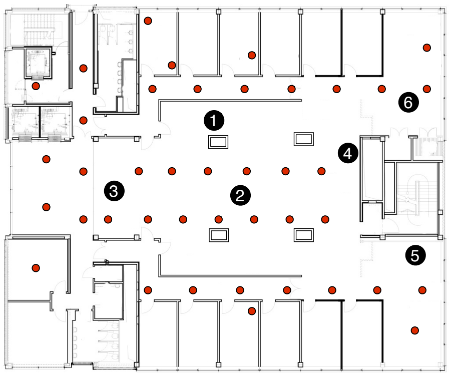


Figure 12: Testbed environment: Soekris clients are marked as small dots, and the AP locations are labelled "1"–"6".

Experimental methodology. We place prototype APs at the locations marked "1"–"6" in our testbed floorplan, shown in Figure 12. The layout shows the basic structure of the office but does not include the numerous cubicle walls also present. We place the 41 clients roughly uniformly over the floorplan, covering areas both near to, and far away from the AP. We put some clients near metal, wood, glass and plastic walls to make our experiments more comprehensive. We also place some clients behind concrete pillars in our office so that the direct path between the AP and client is blocked, making the situation more challenging.

To measure ground truth in the location experiments presented in this section, we used scaled architectural drawings of our building combined with measurements taken from a Fluke 416D laser distance measurement device, which has an accuracy of 1.5 mm over 60 m.

Due to budget constraints, we used one WARP AP, moving it between the different locations marked on the map in Figure 12 and receiving new packets to emulate many APs receiving a transmission simultaneously. This setup does not favor our evaluation of ArrayTrack.

4.1 Static localization accuracy

We first evaluate how accurately AoA pseudospectrum computation without array geometry weighting and reflection path removing localizes clients. This represents the performance ArrayTrack would obtain in a static environment without any client movement, or movement nearby. The curves labeled three APs, four APs, five APs, and six APs in Figure 13 show raw location error computed with Equation 8 across all different AP combinations and all 41 clients. We see that the general trend is that average error decreases with an increasing number of APs. The median error varies from 75 cm for three APs to 26 cm for six APs. The average error varies from 317 cm for three APs to 38 cm for six APs. We show a heatmap combination example in Figure 14 with increasing number of APs.

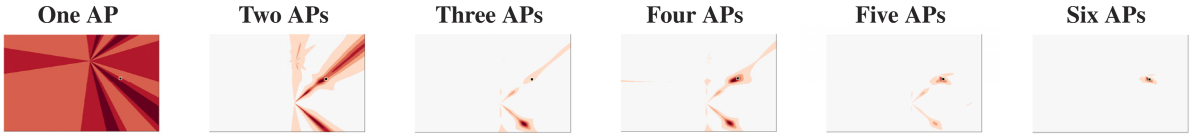


Figure 14: Heatmaps showing the location likelihood of a client with differing numbers of APs computing its location. We denote the ground truth location of the client in each by a small dot in each heatmap.

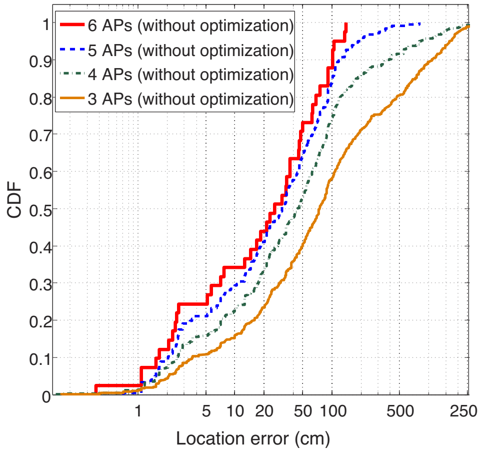


Figure 13: Cumulative distribution of location error from unoptimized raw AoA spectra data across clients using measurements taken at all combinations of three, four, five, and six APs.

4.2 Semi-static localization accuracy

We now evaluate ArrayTrack using data that incorporates small (less than 5 cm) movements of the clients, with two more such location samples per client. This is representative of human movement even when stationary, due to small inadvertent movements, and covers all cases where there is even more movement up to walking speed. In Figure 15, we show that ArrayTrack improves the accuracy level greatly, especially when the number of APs is small. Our system improves mean accuracy level from 38 cm to 31 cm for six APs (a 20% improvement). We measure 90%, 95% and 98% of clients to be within 80 cm, 90 cm and 102 cm respectively of their actual positions. This improvement is mainly due to the array geometry weighting, which removes the relatively inaccurate parts of the spectrum approaching 0 degrees or 180 degree (close to the line of the antenna array).

When there are only three APs, ArrayTrack improves the mean accuracy level from 317 cm to 107 cm, which is around a 200% improvement. The intuition behind this large performance improvement is the effective removal of the false positive locations caused by multipath reflections and redundant symmetrical bearings. When the number of APs is big such as five or six, heatmap combination inherently reinforces the true location and removes false positive locations. However, when the number of APs is small, this reinforcement is not always strong and sometimes the array symmetry causes false positive locations, which greatly degrades the localization performance. In these cases, we enable the array symmetry removal scheme described in Section 2.3.4 to significantly enhance accuracy. By using this technique, ArrayTrack can achieve a median 57 cm accuracy levels with only three APs, good enough for many indoor applications.

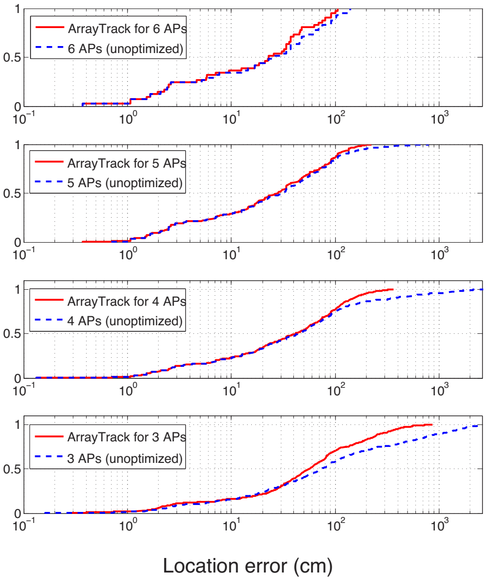


Figure 15: Cumulative distribution of location error across clients for three, four, five and six APs with ArrayTrack.

4.2.1 Varying number of AP antennas

We now show how ArrayTrack performs with differing number of antennas at APs. In general, with more antennas at each AP, we can achieve a more accurate AoA spectrum and capture a higher number of reflection-path bearings, which accordingly increases localization accuracy, as Figure 16 shows. Because we apply spatial smoothing on top of the MUSIC algorithm, the effective number of antennas is actually reduced and so we are not able to capture all the arriving signals when the number of antennas is small. The mean accuracy level is 138 cm for four antennas, 60 cm for six antennas and 31 cm for eight antennas. It's interesting to note that the improvement gap between four and six antennas is bigger than that between six to eight antennas. In a strong multipath indoor environment like our office, the direct path signal is not always the strongest. However, the direct path signal is among the three biggest signals most of the time. We show how the direct path peak changes in Figure 17.

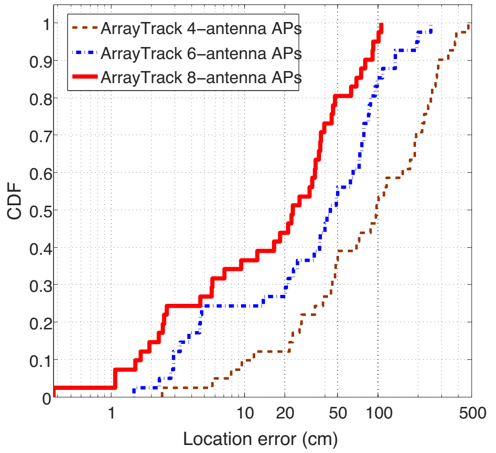


Figure 16: CDF plot of location error for four, six and eight antennas with ArrayTrack.

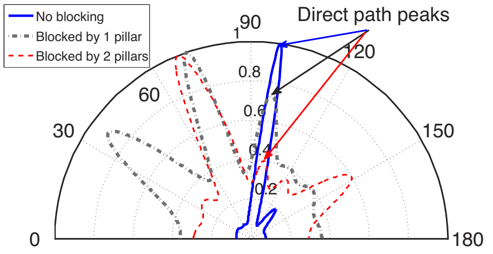


Figure 17: The AoA spectra for 3 clients in a line with AP.

We keep the client on the some line with AP while block it with more pillars. Even when it's blocked by two pillars, the direct path signal is still among the top three biggest, although not the strongest. With five virtual antennas, after spatial smoothing, we are able to avoid losing the direct path signals as sometimes happens when we only use four antennas. The accuracy level improvement from six to eight antennas is due to the more accurate AoA spectrum obtained. With an increasing number of antennas, there will be some point when increasing the number of antennas does not improve accuracy any more as the dominant factor will be the calibration, antenna imperfection, noise, correct alignment of antennas, and even the human measurement errors introduced with laser meters in the experiments. We expect that an antenna array with six antennas (30.5 cm long) or eight antennas (43 cm long) is quite reasonable.

4.3 Robustness

Robustness to varying client height, orientation, low SNR, and collisions is an important characteristic for ArrayTrack to achieve. We investigate ArrayTrack's accuracy under these adverse conditions in this section.

As ArrayTrack works works on any part of the packet. We choose the preamble of the packet to work with ArrayTrack. Preamble part is transmitted at the base rate and what's more, complex conjugate with the known training symbol generate peaks which is very easy to be detected even at low SNR.

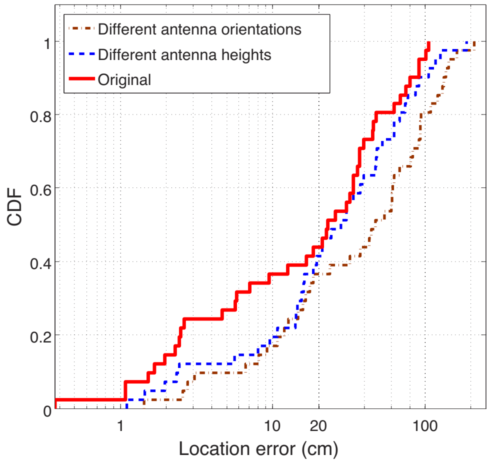


Figure 18: CDF plot of ArrayTrack's location error for different antenna height, different orientation and baseline results, with eight antennas and six APs.

4.3.1 Height of mobile clients

In reality, most mobiles rest on a table or are held in the hand, so they are often located around 1–1.5 meters off the ground. APs are usually located on the wall near the ceiling, typically 2.5 to 3 meters high. We seek to study whether this height difference between clients and APs will cause significant errors in our system's accuracy. The mathematical analysis in §2.3 is based on the assumption that clients and APs are at the same height. In Appendix A we show that a 1.5 meter height difference introduces just 1%–4% error when the distance between the client and AP varies between five and 10 meters. In our experiments, our AP is placed on top of a cart for easy movement with the antennas positioned 1.5 meters above the floor. To emulate a 1.5-meter height difference between AP and clients, we put the clients on the ground at exactly the same location and generate the localization errors with ArrayTrack to compare with the results obtained when they are more or less on the same height with the AP.[4]

The experimental results shown in Figure 18 demonstrate the preceding. Median location error is slightly increased from 23 cm to 26 cm when the AP uses eight antennas. One factor involved is that it is unlikely for a client to be close to all APs, as the APs are separated in space rather than being placed close to each other. One advantage of our system is the independence of each AP from the others, *i.e.*, when we have multiple APs, even if one of them is generating inaccurate results, the rest will not be affected and will mitigate the negative effects of the inaccurate AP by reinforcing the correct location.

[4] Note that this low height does not favor our experimental results as lower AP positions are susceptible to even more clutter from objects than an AP mounted high on the well near the ceiling.

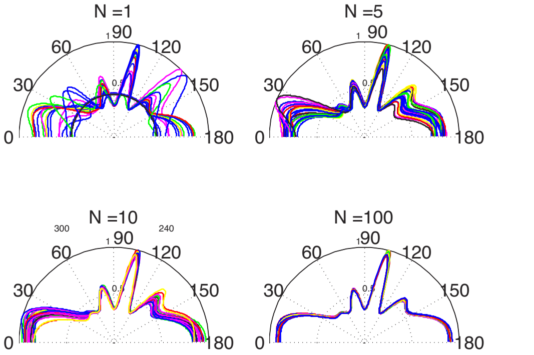


Figure 19: The effect of number of data samples on AoA spectrum.

In future work, we are planning to extend the ArrayTrack system to three dimensions by using a vertically-oriented antenna array in conjunction with the exiting horizontally-oriented array. This will allow the system to estimate elevation directly, and largely avoid this source of error entirely.

4.3.2 *Mobile orientation*

Users carry mobile phones in their hands at constantly-changing orientations, so we study the effect of different antenna orientations on ArrayTrack. Keeping the transmission power the same on the client side, we rotate the clients' antenna orientations perpendicular to the APs' antennas. The results in Figure 18 show that the accuracy level we achieve suffers slightly compared with the original results, median location error increasing from 23 cm to 50 cm. By way of explanation, we find that the received power at the APs is smaller with the changed antenna orientation, because of the different polarization. With linearly polarized antennas, a misalignment of polarization of 45 degrees will degrade the signal up to 3 dB and a misaligned of 90 degrees causes an attenuation of 20 dB or more. By using circularly-polarized antennas at the AP, this issue can be mitigated.

4.3.3 *Number of preamble samples*

To show that ArrayTrack works well with very small number of samples, we present testbed results in Figure 19. Each subplot is composed of 30 AoA spectra from 30 different packets recorded from the same client in a short period of time. We use different number of samples to generate our AoA pseudospectra. As WARPLab samples 40 MHz per second, one sample takes only 0.025 us. We can see that when the number of samples increased to 5, the AoA spectrum is already quite stable which demonstrate ArrayTrack has the potential to responds extremely fast. We employ 10 samples in our experiments and for a 100 ms refreshing interval, the overhead introduced by ArrayTrack traffic is as little as: $\frac{(10\text{ samples})(32\text{ bits/sample})(8\text{ radios})}{100\text{ ms}} = 0.0256$ Mbps.

4.3.4 *Low signal to noise ratio (SNR)*

We show the signal to noise ratio (SNR) effect on the performance of ArrayTrack in this section. Because ArrayTrack does not need to decode any packet content, all the short and long training symbols can be used for packets detection, which performs very well compared with the original Schmidl-Cox packet detection algorithm. With all the 10 short training symbols used, we are able to detect packets at SNR as low as -10 dB.

It's clear that low SNR is not affecting our packet detection at all. Then we want to see whether this low SNR affect our AoA performance. We keep the client at the same position untouched and keep decreasing the transmission power of the client to see how AoA spectra change. The results are shown in Figure 4.3.4. It can be seen clearly that when the SNR becomes very low below 0 dB, the spectrum is not sharp any more and very large side lobe appears on the spectrum generated. This will definitely affect our localization performance. However, we also find that as long as the SNR is not below 0 dB, ArrayTrack works pretty well.

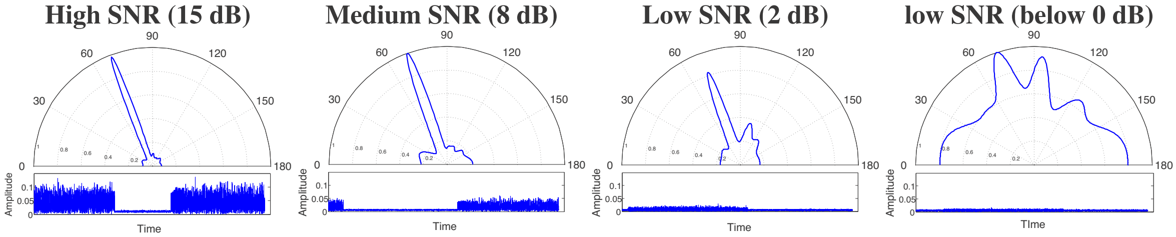
4.3.5 *Packet collisions*

When there are two simultaneous transmissions which causes collision, ArrayTrack still works well as long as the preambles of the two packets are not overlapping.

For collision between two packets of 1000 bytes each, the chance of preamble colliding is 0.6%. We show that as long as the training symbols are not overlapping, we are able to obtain AoA information for both of them using a form of successive interference cancellation. We detect the first colliding packet and generate an AoA spectrum. Then we detect the second colliding packet and generate its AoA spectrum. However, the second AoA spectrum is composed of bearing information for both packets. So we remove the AoA peaks of the first packet from the second AoA spectrum, thus successfully obtaining the AoA information for the second packet.

4.4 System latency

System latency is important for the real-time applications we envision ArrayTrack will enable, such as augmented reality. Figure 21 summarizes the latency our system incurs, starting from the beginning of a frame's preamble as it is received by the ArrayTrack APs. As discussed previously (§4.3.3), ArrayTrack only requires 10 samples from the preamble in order to function. We therefore have the opportunity to begin transferring and processing the AoA information while the remainder of the preamble and the body of the packet is still on the air, as shown in the figure. System latency is comprised of the following pieces:



High SNR (15 dB)

Medium SNR (8 dB)

Low SNR (2 dB)

low SNR (below 0 dB)

Figure 20: AoA spectra become less sharp and more side peaks when the SNR becomes small.

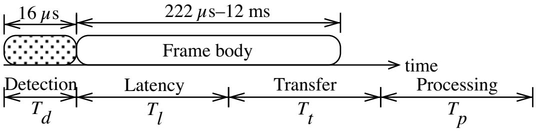


Figure 21: A summary of the end-to-end latency that the ArrayTrack system incurs in determining location.

1. T: the air time of a frame. This varies between approximately 222 μs for a 1500 byte frame at 54 Mbit/s to 12 ms for the same size frame at 1 Mbit/s.

2. T_d: the preamble detection time. For the 10 short and two long training symbols in the preamble, this is 16 μs.

3. T_l: WARP-PC latency to transfer samples. We estimate this to be approximately 30 milliseconds, noting that this can be significantly reduced with better bus connectivity such as PCI Express on platforms such as the Sora [32].

4. T_t: WARP-PC serialization time to transfer samples.

5. T_p, the time to process all recorded samples.

T_t is determined by the number of samples transferred from the WARPs to the PC and the transmission speed of the Ethernet connection. The Ethernet link speed between the WARP and PC is 100 Mbit/s. However, due to the very simple IP stack currently implemented on WARP, added overheads mean that the maximum throughput that can be achieved is about 1 Mbit/s. This yields $T_t = \frac{(10 \text{ samples})(32 \text{ bits/sample})(8 \text{ radios})}{1 \text{ Mbit/s}} = 2.56$ ms.

T_p depends on how the MUSIC algorithm is implemented and the computational capability of the ArrayTrack server. For an eight-antenna array, the MUSIC algorithm involves eigenvalue decomposition and matrix multiplications of linear dimension eight. Because of the small size of these matrices, this process is not the limiting factor in the server-side computations. In the synthesis step (§2.5) we apply a hill climbing algorithm to find the maximum in the heatmap computed from the AoA spectra. For our current Matlab implementation with an Intel Xeon 2.80 GHz CPU and 4 GB of RAM, the average processing time is 100 ms with a variance of 3 ms for the synthesis step.

Therefore, the total latency that ArrayTrack adds starting from the end of the packet (excluding bus latency) is $T_l = T_d + T_t + T_p - T \approx 100$ ms.

5 Related Work

The present paper is based on the ideas sketched in a previous workshop paper [39], but contributes novel diversity synthesis (§2.2) and multipath suppression (§2.4) design techniques and algorithms, as well as providing the first full performance evaluation of our system.

ArrayTrack owes its research vision to early indoor location service systems that propose dedicated infrastructure Active Badge [35] equips mobiles with infrared transmitters and buildings with many infrared receivers. The Bat System [36] uses a matrix of RF-ultrasound receivers, each hard-coded with location, deployed on the ceiling indoors. Cricket [19] equips buildings with combined RF/ultrasound beacons while mobiles carry RF/ultrasound receivers.

Some recent work including CSITE [13] and PinLoc [27] has explored using the OFDM subcarrier channel measurements as unique signatures for security and localization. This requires a large amount of wardriving, and the accuracy is limited to around one meter, while ArrayTrack achieves finer accuracy and eliminates any calibration beforehand.

The most widely used RF information is received signal strength (RSS), usually measured in units of whole decibels. While readily available from commodity WiFi hardware at this granularity, the resulting RSS measurements are very coarse compared to direct physical-layer samples, and so incur an amount of quantization error, especially when few readings are present.

Map-building approaches. There are two main lines of work using RSS; the first, pioneered by RADAR [2, 3] builds "maps" of signal strength to one or more access points, achieving an accuracy on the order of meters [23, 30]. Later systems such as Horus [43] use probabilistic techniques to improve localization accuracy to an average of 0.6 meters when an average of six access points are within range of every location in the wireless LAN coverage area, but require large amounts of calibration. While some work has attempted to reduce the calibration overhead [12], mapping generally requires significant calibration effort. Other map-based work has proposed using overheard GSM signals from nearby towers [34], or dense deployments of desktop clients [4]. Recently, Zee [21] has proposed using crowd-sourced measurements in order to perform the calibration step, resulting in an end-to-end median localization error of three meters when Zee's crowd-sourced data is fed into Horus. In contrast to these map-based techniques, ArrayTrack achieves better accuracy with few APs, and requires no calibration of any kind beforehand, essential if there are not enough people nearby to crowd-source measurements before the RF environment changes.

Model-based approaches. The second line of work using RSS are techniques based on mathematical models. Some of these proposals use RF propagation models [22] to predict distance away from an access point based on signal strength readings. By triangulating and extrapolating using signal strength models, TIX [11] achieves an accuracy of 5.4 meters indoors. Lim *et al.* [14] use a singular value decomposition method combined with RF propagation models to create a signal strength map (overlapping with map-based approaches). They achieve a localization error of about three meters indoors. EZ [8] is a system that uses sporadic GPS fixes on mobiles to bootstrap the localization of many clients indoors. EZ solves these constraints using a genetic algorithm, resulting in a median localization error of between 2–7 meters indoors, without the need for calibration.

Other model-based proposals augment RF propagation models with Bayesian probabilistic models to capture the relationships between different nodes in the network [16], and develop conditions for a set of nodes to be localizable [42]. Still other model-based proposals are targeted towards ad hoc mesh networks [6, 20, 24].

Prior work in AoA. Wong *et al.* [37] investigate the use of AoA and channel impulse response measurements for localization. While they have demonstrated positive results at a very high SNR (60 dB), typical wireless LANs operate at significantly lower SNRs, and the authors stop short of describing a complete system design of how the ideas would integrate with a functioning wireless LAN as ArrayTrack does. Niculescu *et al.* [17] simulate AoA-based localization in an ad hoc mesh network. AoA has also been proposed in CDMA mobile cellular systems [40], in particular as a hybrid approach between TDoA and AoA [9, 38], and also in concert with interference cancellation and ToA [33].

Much other work in AoA uses the technology to solve similar but materially different problems. Geo-fencing [29] utilizes directional antennas and a frame coding approach to control APs' indoor coverage boundary. Patwari *et al.* [18] propose a system that uses the channel impulse response and channel estimates of probe tones to detect when a device has moved, but do not address location. Faria and Cheriton [10] and others [5, 15] have proposed using AoA for location-based security and behavioral fingerprinting in wireless networks. Chen *et al.* [7] investigate *post hoc* calibration for commercial off-the-shelf antenna arrays to enable AoA determination, but do not investigate localization indoors.

6 Discussion

How does ArrayTrack deal with NLOS?

The NLOS encountered in our experiments can be categorized into two different scenarios:

- S1: Direct path signal is not the strongest but exists.
- S2: Direct path signal is totally blocked.

S1 does not affect ArrayTrack as the spectra synthesis method strengthens the true location in nature.

For S2, one blocked direct path degrades the performance of ArrayTrack slightly but not much. It's not very likely the client's direct paths to all the APs are blocked.

Linear versus circular array arrangement?

Most commonly seen commercial APs have their antennas placed in linear arrangement. As circular array resolves 360 degrees while linear resolves 180 degrees, twice the number of antennas is needed for circular array to achieve the same level of resolution accuracy while linear array has the problem of symmetry ambiguity addressed with synthesis of multiple APs. We plan to consider other array arrangements in our future work.

7 Conclusion

We have presented ArrayTrack, an indoor location system that uses angle-of-arrival techniques to locate wireless clients indoors to a level of accuracy previously only attainable with expensive dedicated hardware infrastructure. ArrayTrack combines best of breed algorithms for AoA based direction estimation and spatial smoothing with novel algorithms for suppressing the non-line of sight reflections that occur frequently indoors and synthesizing information from many antennas at the AP.

A AP-Client Height Difference

Suppose the AP is distance h above the client; we compute the resulting percentage error. AoA relies on the distance difference $d_1 - d_2$ between the client and the two AP antennas in a pair. Given an added height, this difference becomes:

$$d_1' - d_2' = \frac{d_1}{\cos\phi} - \frac{d_2}{\cos\phi} \tag{13}$$

where $\cos\phi = h/d$. The percentage error is then $\frac{(d_1'-d_2')-(d_1-d_2)}{d_1-d_2} = (\cos\phi)^{-1} - 1$. For $h = 1.5$ meters and $d = 5$ meters, this is 4% error; for $h = 1.5$ meters and $d = 10$ meters, this is 1% error.

References

[1] E. Aryafar, N. Anand, T. Salonidis, and E. Knightly. Design and experimental evaluation of multi-user beamforming in wireless LANs. In *Proc. of ACM MobiCom*, 2010.

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

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

[4] P. Bahl, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill. DAIR: A framework for managing enterprise wireless networks using desktop infrastructure. In *Proc. of ACM HotNets*, 2005.

[5] S. Bratus, C. Cornelius, D. Kotz, and D. Peebles. Active behavioral fingerprinting of wireless devices. In *Proc. of ACM WiSec*, pages 56–61, Mar. 2008.

[6] S. Capkun, M. Hamdi, and J. Hubaux. GPS-free positioning in mobile ad-hoc networks. In *Proc. of Hawaii Int'l Conference on System Sciences*, 2001.

[7] H. Chen, T. Lin, H. Kung, and Y. Gwon. Determining RF angle of arrival using COTS antenna arrays: a field evaluation. In *Proc. of the MILCOM Conf.*, 2012.

[8] K. Chintalapudi, A. Iyer, and V. Padmanabhan. Indoor localization without the pain. In *Proc. of ACM MobiCom*, 2010.

[9] L. Cong and W. Zhuang. Hybrid TDoA/AoA mobile user location for wideband CDMA cellular systems. *IEEE Trans. on Wireless Communications*, 1 (3):439–447, 2002.

[10] D. Faria and D. Cheriton. No long-term secrets: Location based security in overprovisioned wireless lans. In *Proc. of the ACM HotNets Workshop*, 2004.

[11] Y. Gwon and R. Jain. Error characteristics and calibration-free techniques for wireless LAN-based location estimation. In *ACM MobiWac*, 2004.

[12] A. Haeberlen, E. Flannery, A. Ladd, A. Rudys, D. Wallach, and L. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proc. of ACM MobiCom*, 2004.

[13] Z. Jiang, J. Zhao, X. Li, J. Han, and W. Xi. Rejecting the Attack: Source Authentication for Wi-Fi Management Frames using CSI Information. In *Proc. of IEEE Infocom*, 2013.

[14] H. Lim, C. Kung, J. Hou, and H. Luo. Zero configuration robust indoor localization: Theory and experimentation. In *Proc. of IEEE Infocom*, 2006.

[15] D. C. Loh, C. Y. Cho, C. P. Tan, and R. S. Lee. Identifying unique devices through wireless fingerprinting. In *Proc. of the ACM WiSec Conf.*, pages 46–55, Mar. 2008.

[16] D. Madigan, E. Einahrawy, R. Martin, W. Ju, P. Krishnan, and A. Krishnakumar. Bayesian indoor positioning systems. In *Proc. of IEEE Infocom*, 2005.

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

[18] N. Patwari and S. Kasera. Robust location distinction using temporal link signatures. In *Proc. of the ACM MobiCom Conf.*, pages 111–122, Sept. 2007.

[19] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *Proc. of the ACM MobiCom Conf.*, pages 32–43, Aug. 2000.

[20] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Mobile-assisted localization in wireless sensor networks. In *Proc. of IEEE Infocom*, 2005.

[21] A. Rai, K. Chintalapudi, V. Padmanabhan, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proc. of ACM MobiCom*, 2012.

[22] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice-Hall, 2nd edition, 2002.

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

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

[25] T. Schmidl and D. Cox. Robust Frequency and Timing Synchroniation for OFDM. *IEEE Trans. on Communications*, 45(12):1613–1621, Dec. 1997.

[26] R. Schmidt. Multiple emitter location and signal parameter estimation. IEEE *Trans. on Antennas and Propagation*, AP-34(3):276–280, Mar. 1986.

[27] S. Sen, B. Radunovic, R. Choudhury, and T. Minka. Spot localization using phy layer information. In *Proceedings of ACM MOBISYS*, 2012.

[28] T.-J. Shan, M. Wax, and T. Kailath. On spatial smoothing for direction-of-arrival estimation of coherent signals. *IEEE Trans. on Acoustics, Speech, and Sig. Proc.*, ASSP-33(4):806–811, Aug. 1985.

[29] A. Sheth, S. Seshan, and D. Wetherall. Geo-fencing: Confining Wi-Fi Coverage to Physical Boundaries. In *Proceedings of the 7th International Conference on Pervasive Computing*, 2009.

[30] A. Smailagic, D. Siewiorek, J. Anhalt, D. Kogan, and Y. Wang. Location sensing and privacy in a context aware computing environment. In *Pervasive Computing*, 2001.

[31] K. Tan, H. Liu, J. Fang, W. Wang, J. Zhang,

M. Chen, and G. Voelker. SAM: Enabling practical spatial multiple access in wireless LAN. In *Proc. of ACM MobiCom*, 2009.

[32] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. Voelker. Sora: High performance software radio using general purpose multi-core processors. In *Proc. of the* NSDI *Conf.*, Apr. 2009.

[33] A. Tarighat, N. Khajehnouri, and A. Sayed. Improved wireless location accuracy using antenna arrays and interference cancellation. 4, 2003.

[34] A. Varshavsky, E. Lara, J. Hightower, A. LaMarca, and V. Otsason. GSM indoor localization. In *Pervasive and Mobile Computing*, 2007.

[35] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Trans. on Information Systems*, 10(1):91–102, Jan. 1992.

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

[37] C. Wong, R. Klukas, and G. Messier. Using WLAN infrastructure for angle-of-arrival indoor user location. In *Proc. of the IEEE VTC Conf.*, pages 1–5, Sept. 2008.

[38] Y. Xie, Y. Wang, P. Zhu, and X. You. Grid-search-based hybrd ToA/AoA location techniques for NLOS environments. *IEEE Comms. Letters*, 13 (4):254–256, 2009.

[39] J. Xiong and K. Jamieson. Towards fine-grained radio-based indoor location. In *Proc. of ACM HotMobile*, 2012.

[40] L. Xiong. A selective model to suppress nlos signals in angle-of-arrival (AoA) location esimation. In *Proc. of the IEEE PIMRC*, 1998.

[41] xirrus. Xirrus Corp. (http://www.xirrus.com).

[42] Z. Yang, Y. Liu, and X. Li. Beyond trilateration: On the localizability of wireless ad-hoc networks. In *Proc. of IEEE Infocom*, 2009.

[43] M. Youssef and A. Agrawala. The Horus WLAN location determination system. In *Proc. of ACM MobiSys*, 2005.