



Research Note

RN/11/19

ArrayTrack: A Fine-Grained Indoor Location System

19th October 2011

Jie Xiong

Kyle Jamieson

Abstract

Location systems are key to a rich experience for mobile users. When they roam outdoors, mobiles can usually count on a clear GPS signal for an accurate location, but indoors, GPS usually fades, and so up until recently, mobiles have had to rely mainly on rather coarse grained signal strength readings for location. What has changed this status quo is the recent trend of dramatically increasing numbers of antennas at the indoor AP, mainly to bolster capacity and coverage with multiple-input, multiple-output (MIMO) techniques. In the near future, the number of antennas at the access point will increase several-fold, to meet increasing demands for wireless capacity with MIMO links, spatial division multiplexing, and interference management. 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 in real time as they roam about a building. We prototype ArrayTrack on a WARP platform, emulating the capabilities of an inexpensive 802.11 wireless access point. Our results show that ArrayTrack can pinpoint 40 clients spread out over an indoor office environment to within an 30 cm location accuracy.

ArrayTrack: A Fine-Grained Indoor Location System

UCL CS Research Note RN/11/19

Jie Xiong
University College London
j.xiong@cs.ucl.ac.uk

Kyle Jamieson
University College London
k.jamieson@cs.ucl.ac.uk

Location systems are key to a rich experience for mobile users. When they roam outdoors, mobiles can usually count on a clear GPS signal for an accurate location, but indoors, GPS usually fades, and so up until recently, mobiles have had to rely mainly on rather coarse-grained signal strength readings for location. What has changed this status quo is the recent trend of dramatically increasing numbers of antennas at the indoor AP, mainly to bolster capacity and coverage with multiple-input, multiple-output (MIMO) techniques. In the near future, the number of antennas at the access point will increase several-fold, to meet increasing demands for wireless capacity with MIMO links, spatial division multiplexing, and interference management. 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 in real time as they roam about a building. We prototype ArrayTrack on a WARP platform, emulating the capabilities of an inexpensive 802.11 wireless access point. Our results show that ArrayTrack can pinpoint 40 clients spread out over an indoor office environment to within an 30 cm location accuracy.

1. INTRODUCTION

The proliferation of mobile computing devices continues today, with smartphones, tablets, and laptops a commonplace sight. Outdoors, mobile 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, two factors make providing an accurate location service quite challenging. First, the many objects found indoors near access points and mobile clients reflect the energy of the wireless signal in a phenomenon called multipath propagation. While the differences between the resulting path lengths vary over a wide range outdoors (on the order of $3 \mu\text{s}$ in urban areas), they vary over a significantly smaller range indoors (about 20–50 ns) [17]. This forces an unfortunate tradeoff that most existing

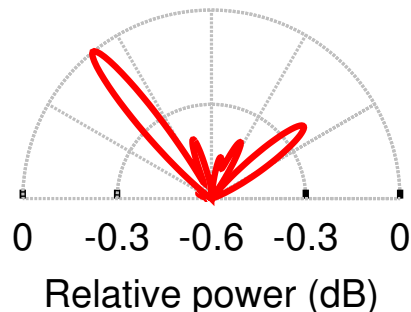


Figure 1: A *pseudospectrum* of a client's received signal at a multi-antenna access point estimates the incoming signal's power as a function of its angle of arrival.

location-based systems make: either adapt to 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 systems choose the former, building maps of multipath signal strength [4, 5, 32, 25], or estimating coarse differences using RF propagation models [9, 11].

The second factor that makes an indoor location service challenging is that users' and applications' demands for accuracy are especially acute indoors. While the few meters of accuracy GPS provides outdoors are more than sufficient for street- or city block-level navigation, small differences in location have more meaning to people and applications indoors. A few meters of error in an estimated location can place a person in a different room within a building, for example. A solution that offers a centimeter-accurate location service could therefore allow widely-deployed WiFi access points to offer the benefits of systems that heretofore required dedicated infrastructure, such as Active Badge [26], Bat [27], and Cricket [16].

The key observation we make in this paper is that in recent years, a new opportunity to improve indoor location systems has presented itself: an ever-increasing number of antennas at the access point, mainly to bolster capacity and coverage with multiple-input, multiple-output (MIMO) techniques. IEEE 802.11n, in particu-

lar, exploits MIMO extensively through the use of many antennas at the access point. 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 division multiplexing [23, 1].

ArrayTrack is a system that exploits the increasing number of antennas at commodity access points to provide fine-grained location for mobile clients in an indoor setting. To make ArrayTrack feasible in terms of cost, we use commodity hardware that samples the wireless signal at 802.11 bandwidth (20 MHz). Two or more ArrayTrack APs cooperate to use angle-of-arrival (AoA) of clients’ incoming signals at each to determine the client’s location. The result, shown in Figure 1, is a *pseudospectrum*: a high-resolution estimate of power arriving at the AP as a function of angle. The specific challenge that multipath poses is the addition of side-lobes in the pseudospectrum corresponding to reflections at angles other than the client’s true bearing. To address this problem, we introduce a novel architecture and multipath disambiguation algorithm that intelligently switches between two heuristics for analyzing the AoA pseudospectrum. Our architecture relies on an AP design with two sets of antennas, separated by approximately two and a half feet.

ArrayTrack advances the known state-of-the-art in AoA-based localization by combining two novel heuristics with “best of breed” algorithms for AoA-based direction finding and dealing with coherent signals arriving at the access point (spatial smoothing), as is common in indoor environments. The first heuristic we propose, *intra-AP triangulation*, uses triangulation between these two sets of antennas to disambiguate the direct path to the client from multipath reflections and simultaneously solve for the client’s true bearing and range. The second heuristic we propose, *pseudospectrum matching*, correlates individual lobes from the two pseudospectra with one another in an attempt to eliminate multipath reflections from each.

A key feature of our approach is that by operating within the physical layer, we can estimate location based on overhearing little more than a packet’s preamble. This allows ArrayTrack to determine a client’s location to within centimeters in real time, something not possible with model- or map-based approaches, which build up information by averaging over relatively long time windows, resulting in median location errors between two and four meters [7].

We implement ArrayTrack on the Rice WARP FPGA platform, and evaluate in a 40-node wireless network deployed over one floor of a busy office space. For all of the stationary clients we tested, intra-AP triangulation and pseudospectrum matching can consistently localize to an angular accuracy of 1.3° , corresponding to an

average linear accuracy of 30 cm for an average 15 m distance between access point and clients.

2. DESIGN

We now describe ArrayTrack’s design top-down, dividing into three different modules.

2.1 Packet detection

To obtain a bearing information for a client, the AP needs to overhear some transmissions from this client. The transmission can be in any form of packet: RTS/CTS or data packet. ArrayTrack only needs a very tiny part of the packet to process the AoA information. ArrayTrack works with any part of the packet and it does not require decoding the packet. For a wireless transmitted packet, the most robust part is the preamble as it’s normally transmitted at base rate and further more, the preamble part contains the known time domain sequence for the receiver to detect the existence of a packet. So ArrayTrack detects the preamble of the packet and records a small part of it. Principally speaking, one time domain packet sample (a 1000-byte packet transmitted at 6 Mbps sampled at 40 MHz contains around 50000 samples) will work for our scheme. However, the packet recorded will be affected by background noise and interference from other senders. We therefore capture multiple samples to obtain mean phase difference. For a 1000-bytes packet, a small part of the preamble such as 1 byte is enough for ArrayTrack to work well. The preamble part of 801.11a/g packets contains known short and long training symbols. We implemented modified version of Schmidl-Cox algorithm on WARP FPGA to detect the short training symbols. As there are 10 short training symbols in the preamble, we apply a moving average filter with window size equal to 10 short training symbols to enhance the detection of the packet. Once a packet is detected, multiple samples of the packet are recorded to process AoA spectrum for this transmission.

2.2 AoA spectrum generation

In both indoor and outdoor wireless channels, a sender’s signal reflects off objects in the environment, resulting in multiple copies of the signal arriving at the access point; this phenomenon is known as multipath. For clarity of exposition, we first describe how to compute angle of arrival when there is just one path from transmitter to access point, then generalize the principles to handle multipath wireless propagation. The key to computing angle of arrival of a wireless signal is to analyze its phase, a quantity that progresses linearly from zero to 2π every radio wavelength λ along the path from client to access point, as shown in Figure 2(a).

This means that the access point receives signals with an added phase determined by the path length d from

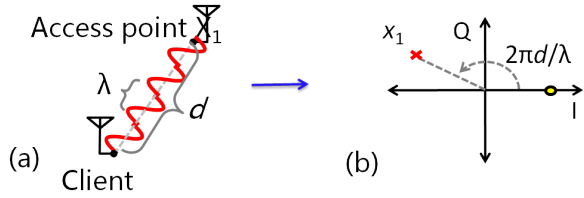


Figure 2: ArrayTrack's principle of operation: (a) The phase of the signal goes through a 2π cycle every radio wavelength λ . (b) The complex representation of the sent (filled dot) and received (crosses) signals at the antenna in (a).

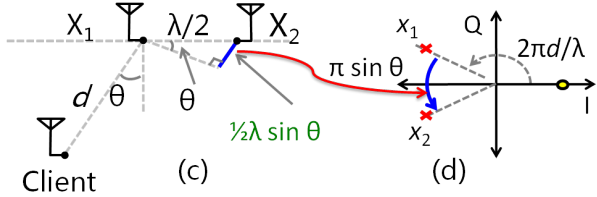


Figure 3: (c) A signal arriving at bearing θ to two antennas. (d) The complex representation of the sent (filled dot) and received (crosses) signals at both antennas in (c).

the client. Phase is particularly easy to analyze 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 2(b), where angle measured from the I axis indicates phase. Using the I-Q plot, we see that the distance d adds a phase of $2\pi d/\lambda$ as shown by the angle measured from the I axis to the cross labeled x_1 (representing the signal received at antenna one). While there are two antennas attached to one AP and located with $\lambda/2$ distance in between, the distances from client to each of the antenna are not equal unless the bearing θ is equal to 0. As depicted in Figure 3(c), 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, the fraction depending on θ . Assume the distance d is much larger than $\lambda/2$ and apply simple mathematics, this amount of extra distance is calculated as:

$$\Delta d = \lambda/2 * \sin \theta \quad (1)$$

These facts suggest a particularly simple way to compute θ at a two-antenna access point in the absence of multipath. First, use a software-defined or hardware radio to measure x_1 and x_2 directly, compute the phase of each ($\angle x_1$ and $\angle x_2$), and then solve for θ ($\angle x_1 - \angle x_2$ is between $-\pi$ and π) as:

$$\theta = \arcsin \left(\frac{\angle x_2 - \angle x_1}{\pi} \right) \quad (2)$$

In real-world multipath environments, however, Equation 2 breaks down because multiple paths' signals sum in the I-Q plot, breaking the simple two-antenna exposition above. However, adding antennas can resolve the ambiguity. We applied the MUSIC algorithm which is the well known AoA estimation algorithm for our estimation. MUSIC algorithm is able to provide unbiased estimates of the number of signals, the angle of arrival, and the strengths of the waveforms. We assume D signals are coming at M antennas and the number of incoming signals D is smaller than the number of antennas M . It's understood that the signals are time varying and thus our calculations are based upon time snapshots of the incoming signal. D signals are arriving from D directions and are received by an array of M antennas. Assume $s_i(k)$ is the i th incoming signal at time k while $x_j(k)$ is the received signal at j th antenna element at time k , then all the D incoming signals are represented as a vector $s(k)$

$$s(k) = \begin{bmatrix} s_1(k) \\ s_2(k) \\ \dots \\ s_D(k) \end{bmatrix} \quad (3)$$

while all the received signals at M antennas are represented by a vector $x(k)$

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \dots \\ x_M(k) \end{bmatrix} \quad (4)$$

If a signal is coming at a bearing θ_i , assume the distance between the signal source and the nearest antenna is d , then the distance between the source and the second antenna d_2 is calculated as:

$$d + 1/2\lambda \cos(\theta) \quad (5)$$

The distance for the M th antenna d_M can easily be found the same way:

$$d + (M - 1)/2\lambda \cos(\theta) \quad (6)$$

Recall the phase change and distance prorogation relationship discussed earlier, $a(\theta_i)$ is defined as the M -element array steering vector for the θ_i direction of signal arrival:

$$a(\theta_i) = a \exp\left(\frac{-j2\pi d}{\lambda}\right) \begin{bmatrix} 1 \\ \exp(-j\pi \lambda \cos(\theta_i)) \\ \exp(-j2\pi \lambda \cos(\theta_i)) \\ \dots \\ \exp(-j(M-1)\pi \lambda \cos(\theta_i)) \end{bmatrix} \quad (7)$$

For D incoming signals, the steering matrix A is defined

as:

$$A = [a(\theta_1) \ a(\theta_2) \ \dots \ a(\theta_D)] \quad (8)$$

By multiplying the incoming signals $s(k)$ with this steering matrix A and adding the noise, the received signals at M antennas $x(k)$ is generated in the following form:

$$x(k) = [a(\theta_1) \ a(\theta_2) \ \dots \ a(\theta_D)] \begin{bmatrix} s_1(k) \\ s_2(k) \\ \dots \\ s_D(k) \end{bmatrix} + n(k) = A * s(k) + n(k) \quad (9)$$

$n(k)$ is the noise vector at the antenna array. It has zero mean and σ_n^2 variance.

Now let us define the $M \times M$ array correlation matrix R_{xx} :

$$R_{xx} = E[x * x^H] \quad (10)$$

$$= E[(As + n)(s^H A^H + n^H)] \quad (11)$$

$$= AE[s * s^H]A^H + E[n * n^H] \quad (12)$$

$$= AR_{ss}A^H + R_{nn} \quad (13)$$

This correlation matrix can also be expressed as:

$$R_{xx} = A \times R_{ss} \times A^H + \sigma_n^2 I \quad (14)$$

where R_{ss} is the source correlation matrix and R_{nn} is the noise correlation matrix. Then the eigenvalues and eigenvectors for R_{xx} are calculated. The array correlation matrix has M eigenvalues $[\lambda_1 \ \lambda_2 \ \dots \ \lambda_M]$ along with M eigenvectors $E = [e_1 \ e_2 \ \dots \ e_M]$. If the eigenvalues are sorted from smallest to the largest, the smallest eigenvalues correspond to the noise while the bigger ones correspond to the signals. Then the matrix E can be divided in to two subspaces as: $E = [E_N \ E_S]$. The first subspace E_N is called the noise subspace and is composed of $M - D$ eigenvectors while the second subspace E_S is called the signal subspace and is composed of D eigenvectors associated with the D arriving signals. For uncorrelated noise, the smallest eigenvalues $\lambda_1 = \lambda_2 = \dots = \lambda_{M-D} = \sigma^2$ correspond to the noise level. In practice, these multiple small eigenvalues will occur in a cluster rather than all precisely equal. This can actually help us to identify the number of incoming signals by finding the number of clustered small eigenvalues. These eigenvectors with the smallest eigenvalues are chosen to construct the $M \times (M - D)$ dimensional noise subspace:

$$E_N = [e_1 \ e_2 \ e_3 \ \dots \ e_{M-D}] \quad (15)$$

The noise subspace eigenvectors are orthogonal to the array steering vectors at the angles of arrival $\theta_1, \theta_2, \dots, \theta_D$. Because of this orthogonal condition, the Euclidean distance $d^2 = a(\theta)^H E_N E_N^H a(\theta) = 0$ holds for each arrival angle. We define MUSIC pseudospectrum

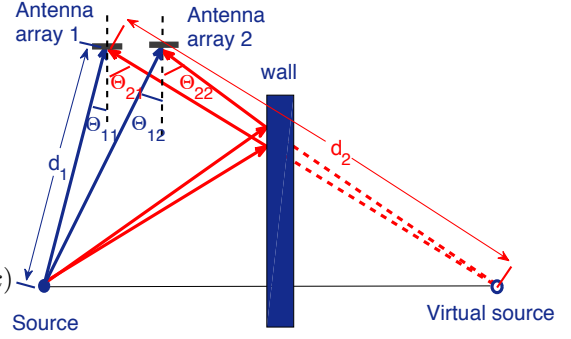


Figure 4: Identify direct path bearing by distance comparison: $d_1 < d_2 \Rightarrow \theta_{11}$ is direct path bearing.

as:

$$P_{MU} = \frac{1}{a(\theta) * E_N * E_N^H * a(\theta)} \quad (16)$$

This expression $a(\theta)^H E_N E_N^H a(\theta)$ will not be exactly 0 in practice but a very small value. Put this expression in the denominator will get sharp peaks at the angles of arrival. When the incoming signals are highly correlated, the traditional MUSIC algorithm does not work well and a modified version of MUSIC with spatial smoothing technique [20] will be applied. For the direct path signal and reflection path signal from the same source, they are highly correlated so we implement spatial smoothing MUSIC for our AoA estimation.

2.3 Direct and reflection paths differentiation scheme

MUSIC algorithm helps us obtain the AoA spectrum for the incoming signals. However, if there are multiple peaks on the spectrum indicating several simultaneous incoming signals from different directions including one direct path and several reflection paths, it's critical for us to identify the direct path bearing for localization. One obvious way to identify the direct path is by measuring which signal arrives first. The channel impulse response (CIR) scheme can be applied to show the incoming signals at different time points. However, CIR scheme requires a very high sampling frequency to differentiate the direct path and reflection path signals. For 802.11a/g with 20 MHz bandwidth, normally the incoming signal is sampled at 40 MHz. This frequency is far too low for CIR scheme to work. So CIR method works well for UWB but not WiFi signals in indoor environments. We propose a novel scheme to differentiate the direct path and reflection paths with two antenna arrays. In the future with more antennas attached to one AP, the antennas can easily be divided into two groups and separated with a small distance. The basic idea of our scheme is demonstrated in Figure 4. Two antenna arrays are placed close to each other in one line.

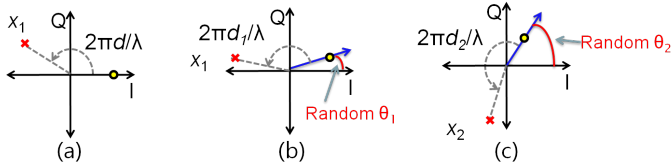


Figure 5: WARPs introduce random but constant phase offsets to each antenna. (a) No phase offset. (b) Random phase offset θ_1 introduced at antenna 1. (c) Random phase offset θ_2 introduced at antenna 2.

The direct path signal arrives at the two arrays with different bearings. As the two arrays are put close to each other, these two bearings also have similar values. We identify two bearings as one pair if they are close enough by comparing the difference with a threshold. The same thing happens to the reflection path. The signal from virtual source indicated in Figure 4 arrives at the two arrays with another two different but close bearings. By knowing these two pairs of bearings, we are able to obtain d_1 and d_2 indicated in Figure 4 respectively:

$$d_1 = \left| \frac{1}{\tan \theta_{11} - \tan \theta_{12}} * \frac{1}{\cos \theta_{11}} \right| \quad (17)$$

$$d_2 = \left| \frac{1}{\tan \theta_{21} - \tan \theta_{22}} * \frac{1}{\cos \theta_{21}} \right| \quad (18)$$

As we know, the direct path distance is always the shortest. d_1 and d_2 are compared to identify the direct path bearing. If there are more than two pairs, the minimum distance pair is chosen. In real indoor environment shown later in the evaluation part, we show that it's even simpler for most of the cases because the direct path bearing will change slightly while the reflection path bearings change significantly. So a lot of time, we can only find the direct path bearing pair and it can be easily chosen without calculation of distance and comparison.

3. IMPLEMENTATION CHALLENGES

For the hardware implementation, some challenges exit before our system fully function. We describe them below:

3.1 Antenna phase calibration

Equipping the access point with multiple antennas is necessary for ArrayTrack, but does not suffice to calculate angle of arrival as described in the preceding section. As we see in the right-hand section of Figure 6 labeled “WARP,” each radio receiver incorporates a 2.4 GHz oscillator whose purpose is to convert the incoming radio frequency signal to its representation in

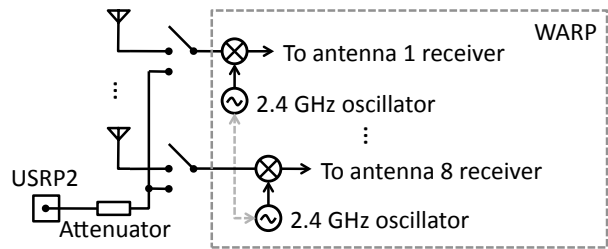


Figure 6: ArrayTrack physical layer design. All 2.4 GHz oscillators are synchronized, and inputs to the eight receivers switch between the antennas and the signal generator (labeled “USRP2”); we use the latter for calibration.

I-Q space shown, for example, above in Figure 5. An undesirable consequence of this *downconversion* step is that it introduces an unknown phase offset to the resulting signal in I-Q space, rendering our proposed method of measuring angle of arrival (and MIMO) inoperable. To remedy this, MIMO systems *phase lock* each radio’s oscillator together, so that they run at exactly the same frequency. We represent this by the dotted line between oscillators in Figure 6. This suffices for MIMO, but not for our application, because the downconverters of even phase-locked systems introduce an unknown but constant phase difference to each receiver, which manifests as an unknown phase added to the constellation points in Figure 5. Our solution is to calibrate the array, measuring each phase offset directly. The USRP2 in Figure 6 transmits a continuous 2.4 GHz carrier through a 36 dB attenuator, which we split into eight signals and feed into the radio front ends. Since each of the eight paths from the USRP2 to a radio receiver is of equal length, the signals we measure when the switches in Figure 6 are each in the lower position yield seven relative phase offsets for antennas 2–8, relative to antenna one. Subtracting these relative phase offsets from the incoming signals over the air then cancels the unknown phase difference, and our scheme become applicable.

3.2 WARP antenna fine time synchronization

As ArrayTrack relies on the phase difference between antennas to obtain AoA information, ArrayTrack fails to achieve correct bearing information if the data samples recorded at each antenna are not fully time synchronized. The signals arrive at each antenna are not the same as each antenna is located at a different positions which experience different multipaths and shadowing. Thus if we apply Schmid-Cox algorithm on each antenna and trigger the “start capturing” when packets are detected, the radios connected to the different antennas may start recording packets at different time points. This problem is solved by selecting one master radio board and only implementing Schmid-Cox algorithm on this board. The rest radios share the

the decision made by the chosen representative radio board. Sharing this decision between the 4 radios on the same WARP board can be accomplished by modifying the hardware system generator model design. However, sharing this decision between two WARPs is more challenging. We utilize the digital I/O pins on the WARP board to output this one bit boolean signal to the other WARP board. The hardware design is modified and two I/O pins are connected with copper wire (100-mil header) to share this decision information.

3.3 Real-time system

The current WARPLab reference design utilizes a custom version of xilnet, a very simple IP stack from Xilinx which the throughput performance is not optimized. The link is 100M Ethernet but the actual throughput is much smaller. The original design records everything in a time window no matter there is packet transmission or nothing but noise. So the transfer of the recorded content from WARP to PC in real time is not possible as the recording speed is much faster than the transferring speed. So the current WARPLab design fills the buffer first before transferring buffer content to the PC which introduce an additional 0.4 ms latency delay. What make this situation worse: the buffer size of WARP is very small that WARP is only able to buffer 16383 samples at 40 MHz sampling rate which corresponds to 0.4 ms. This time period is too short and can only accommodate one small size packet. We implement the Schimdl-Cox algorithm on FPGA to detect the packet and only start recording when there is network traffic going on. This partially solves the problem as we ignore those samples with no meaningful content but noise. Further, we buffer only a small part of the preamble for each packet which reduce the amount of samples needed to be transferred to PC significantly. With these two schemes applied, the time latency is significantly reduced and we are able to buffer and transmit nearly at the same time to achieve a real-time AoA tracking system.

4. EVALUATION

In this section, testbed results are presented to show how ArrayTrack performs in real indoor environment. First we present how accurately ArrayTrack can obtain the direct path bearing of the clients and identify in what circumstances ArrayTrack may fail. We then explore the robustness of ArrayTrack against collision and low SNR. Then we show the latency introduced by ArrayTrack which is a critical factor for a fast response real-time functional system. Finally we present the effect of number of antennas for ArrayTrack and also the spatial smoothing effect on coherent signals in real indoor environment.

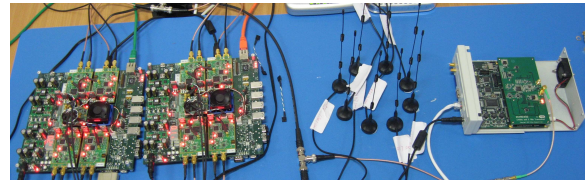


Figure 7: The ArrayTrack prototype. Two Rice WARP platforms (left) provide a total of eight antennas and radio chains, while a cable-connected USRP2 software-defined radio (right) calibrates the array.

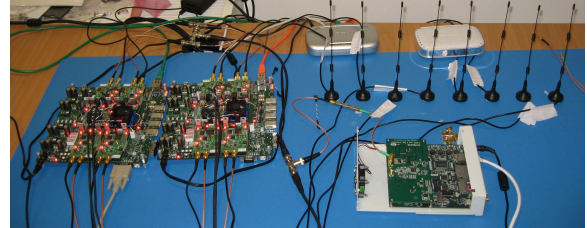


Figure 8: The ArrayTrack prototype with linear antenna arrangement.

Our prototype AP as shown in Figure 7 uses two Rice WARP FPGA-based wireless platforms. Each WARP platform is equipped with four radio front ends and four omnidirectional antennas. The WARPs run a custom hardware design of WARPLab. All the eight antennas are calibrated with respect to one of them as described earlier. The clients are not equipped with multiple antennas. An ordinary laptop with WiFi interface serves perfectly as a client working with our ArrayTrack implemented AP. We use the Soekris boxes as our clients as they are small and can be powered with Ethernet cable which are flexible to be put at any location without the constraint of power plug. Each Soekris box has two antennas and only one antenna transmits while the other is disabled.

The 8 antennas attached to WARP are placed in linear (Figure 8) or circular (Figure 7) arrangements. The AoA range for the linear arrangement is between -90 and 90 degrees, since clients on the two sides of the line formed by the antennas are not differentiable. In the linear arrangement, antennas are spaced at a half wavelength distance (6.13 cm). The circular arrangement is actually an octagon with 4.7 cm sides and an antenna at each corner. We place the prototype access point at the point marked “AP” in our testbed floorplan, shown in Figure 9. The layout shows the basic structure of the office but does not include the cubic layout. It’s a typical indoor office environment commonly seen. We randomly place the 40 soekris clients to cover all the different kinds of scenarios: some of the clients are put far away from the AP and some are

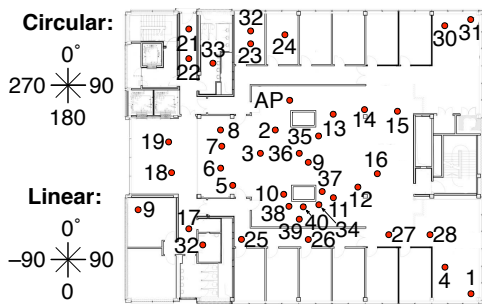


Figure 9: Testbed environment: Soekris clients are numbered, and the WARP access point is labeled “AP.”

placed in other offices with walls between the clients and the AP. We also intentionally put some clients behind the pillar so the direct path between the AP and client is blocked and make the situation more challenging for ArrayTrack. We test ArrayTrack with all possible scenarios to obtain comprehensive results.

4.1 Indoor localization: bearing accuracy

In order to provide accurate localization service in indoor environment, the AoA bearing obtained should be as accurate as possible. We examine pseudospectra from the 40 Soekris clients shown in Figure 9. In an indoor environment with strong multipath propagation, reflections may generate false positive direct path AoA results. It is critical for us to eliminate these false positive AoAs if we are to determine the true bearing of clients. Without any differentiation scheme, we compute the bearing of each client as the angle corresponding to the maximum point on its pseudospectrum. We compute 30 pseudospectra for each client, each from a different packet recorded at different time, and plot the mean obtained bearing as well as 99% confidence interval in Figure 10.

The x axis represents the true bearing we measure from the map. The y axis is the bearing we obtain from ArrayTrack without the proposed differentiation scheme applied. If the mean value is close to the blue line, it means the bearing obtained is accurate and close to the true bearing. From the figure, it can be seen that we are able to obtain accurate bearing results for most clients regardless of proximity to the AP or location inside or outside the AP’s room. For some of them, we get false positive results which means the maximum point corresponds to the reflection path bearing and not the direct path bearing. However, we find this kind of scenarios do not happen often. When clients are located in other rooms, both the direct path and reflection path are attenuated by the walls. So most of the time, the direct path is still detected stronger. The stronger reflection path normally happens when the clients and AP

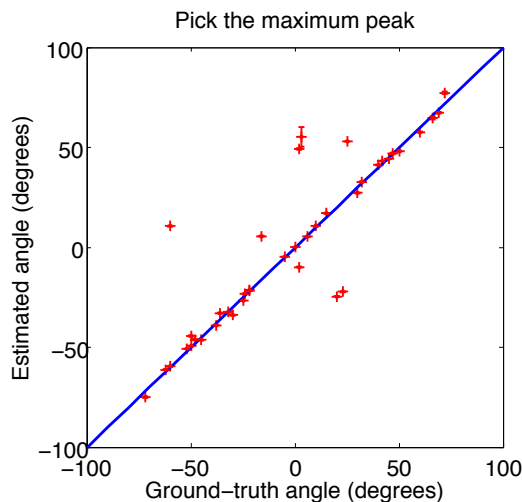


Figure 10: Measured versus ground truth bearing estimation for clients in the office environment (error bars indicate 99% confidence intervals) without path differentiation scheme.

are in the same room and LOS is blocked by obstacles. So the challenging part is to differentiate the direct path and reflection paths for these scenarios.

By applying our differentiation scheme proposed, the results are presented in Figure 11: only one client’s bearing is not detected correctly; the bearings of all other 39 clients are estimated very accurately with a mean error of 1.3 degree. The 99% confidence show the stableness of our scheme with time. This shows the effectiveness of our differentiation scheme in identifying the direct path bearing.

Now let’s zoom into those clients detected wrongly without differentiation scheme and detected correctly with our proposed scheme. For all the 7 clients, they fall into two groups:

Group 1: One reflection path is stronger in at least one group of array and sometimes in both. However, with two groups of arrays, the reflection path bearing are very different while the direct path bearings are more less stable although they may not be the maximum peak in the spectrum. This is shown by one real measurement example from the tested in Figure 12. 5 clients fall into this category.

Group 2: the reflection path and direct path bearings in two arrays form several pairs as described earlier in the design session. We need to calculate the distance and choose the minimum one which corresponds to the direct path bearing. There are two clients belong to this category. One typical example is shown in Figure 13.

There is one client (client 39 in Figure 9) which our scheme fails. There are two pillars in-between client 39 and the AP. ArrayTrack is able to differentiate the

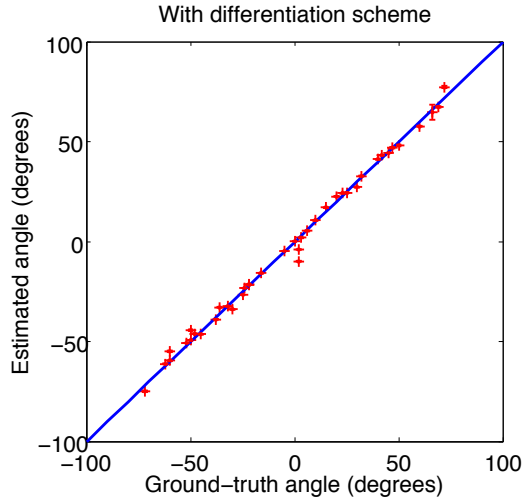


Figure 11: Measured versus ground truth bearing estimation for Soekris clients in the office environment with path differentiation scheme.

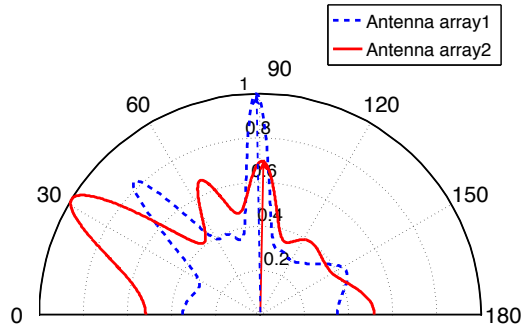


Figure 12: The direct path bearing is stable and the reflection bearings change significantly at two arrays.

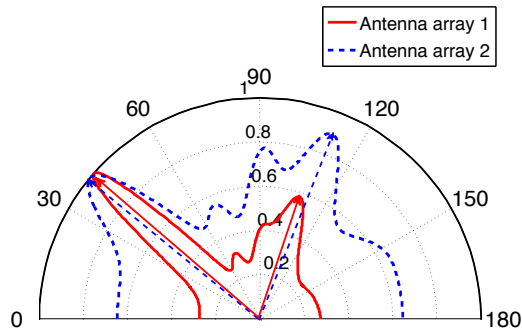


Figure 13: Multiple bearing pairs exist and the bearing corresponds to the minimum distance is chosen.

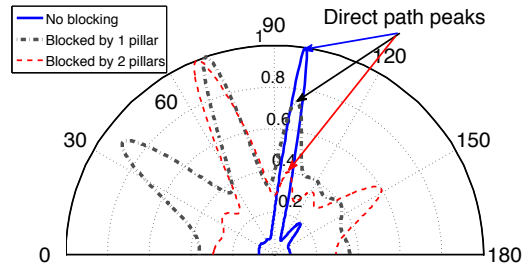


Figure 14: The AoA spectra for 3 clients in a line with AP.

direct path and reflection path efficiently. However, in some extreme case when the direct path is blocked by metal or in our scenario here, blocked by two big pillars, ArrayTrack may fail. To fully understand this, let's see the effect of obstacles blocking the AP and client on our AoA spectrum as shown in Figure 14:

In Figure 14, we choose three positions in a line relative to the AP. The first position is chosen to have no obstacle in-between with the AP. The second position is blocked by one pillar and the third position is behind two pillars. We can clearly see the decreasing of direct path bearing peak in the spectrum graph. However, it should be noted that not all the positions behind the pillar will have reflection path stronger than direct path.

As the client which ArrayTrack fails to detect the direct path bearing is blocked by two pillars, we choose 8 other positions near the failed client which are also blocked by two pillars to test whether ArrayTrack may also fail in these positions. We choose 4 positions horizontally with 20 cm distance in-between and 4 positions vertically. Among these 8 selected positions, ArrayTrack only fails with one position and still succeed with all the rest 7 as the direct path is not totally blocked in these positions. So ArrayTrack more or less works all the time in indoor environment even with strong multipaths except in some very extremely scenarios when the direct path is totally blocked and this is rare in real indoor environment.

4.2 Robustness

Robustness is one important characteristics ArrayTrack wants to achieve. As ArrayTrack works with the preamble part of a packet, ArrayTrack will be robust against low SNR. Preamble part is transmitted at the base rate and what's more, complex conjugate with the know training symbol generate peaks which is very easy to be detected even at low SNR. When there are two simultaneous transmissions which causes collision, ArrayTrack still works well as long as the first several short training symbols of the two packets are not collided. For two packets with A and B bytes receptively, if collision happens, the percentage of these training symbols (assume we utilize all the 10 short training symbols (6

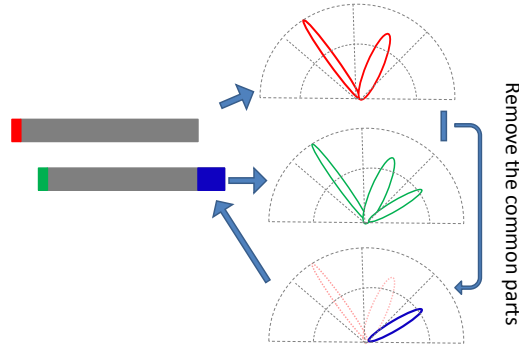


Figure 15: The procedure to obtain AoA spectra for two colliding packets.

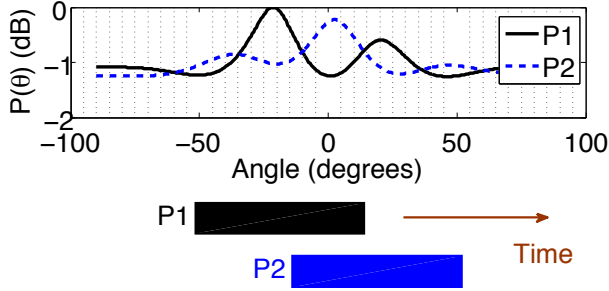


Figure 16: ArrayTrack is able to obtain AoA for two colliding packets.

bytes) to detect the packet) part get collided is calculated as below:

$$\frac{12}{A+B} \quad (19)$$

If both packets are 1000 bytes. The chance of a colliding which will affect ArrayTrack will be 0.6% which is pretty small. If we only utilize one short training symbol to detect the packet, this percentage will further decrease. We show that as long as the training symbols are not colliding, we are able to obtain AoA information for both of them iteratively as shown in Figure 15. The first colliding packet is detected and AoA spectrum is generated. Then the second packet is detected and AoA spectrum is generated. However, the second AoA spectrum is composed of bearing information for both packets as it can be considered as more signals coming from different directions at the same time. Then we remove the AoA peak of the first packet from the second AoA spectrum and we successfully obtain the AoA information for the second packet.

In order for this collision to happen, we intentionally turned off CSMA on Sokeris so two clients can transmit simultaneously. By applying the scheme described above, we are able to obtain AoA information for both the packets as shown in Figure 16.

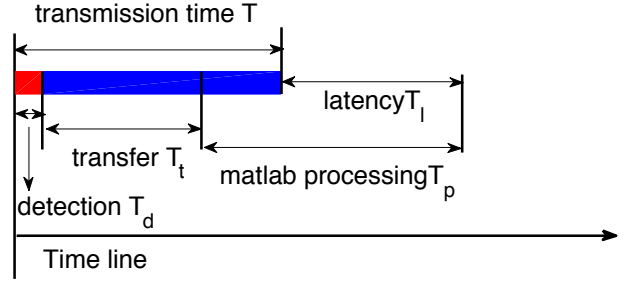


Figure 17: The latency ArrayTrack system introduces.

4.3 System latency

The latency of TrrayTrack is an important characteristic we pay attention to as it's critical for ArrayTrack to work in real time. As ArrayTrack only requires a small part of packet to process AoA information, we are given the opportunity to start transferring and processing the AoA information while the packet is still under transmission. The time-line is shown as Figure 17. ArrayTrack has three parts of latency components:

T : transmission time of a packet.

T_d : the short training symbol detection time. For 10 short training symbols. This time is $0.8 * 10 = 8$ us

T_t : the transfer of the recorded samples from WARP boards to PC through the Ethernet cable. This time is decided by the number of samples transferred from WARP boards to PC and the transmission speed of the Ethernet connection. Assume N samples are to be transmitted and the Ethernet speed is S Mbps. Each sample is 4 bytes and we have 8 radio boards to transfer one by one. So the time needed for the transmission is calculated as: $T_t = N * 32 * 8 / (1000 * S)$ ms

The Ethernet connection between WARP and PC is 100 Mbps. However, due to the very simple IP stack currently implemented on WARP, the maximum throughput can be achieved is much smaller and measured to be around 1 Mbps. To show that ArrayTrack works well with very small number of samples, we present one typical testbed results in Figure 18. Each subplot is composed of AoA spectra from 30 different packets recorded at different time from the same client. We can see clearly that when the number of samples increased to 5, the AoA spectrum is already quite stable which demonstrate ArrayTrack has the potential to respond extremely fast.

Assume we use 5 ($N=5$) samples for each process.

$$T_t = 5 * 32 * 8 / 1000 = 1.28 \text{ ms}$$

T_p : the precessing of recorded samples with Matlab.

T_p depends how the MUSIC algorithm is implemented and the computer capability. For our current implementation with an Intel Xeon 2.80 GHz CPU and 6 G RAM, the processing time is around 5 ms.

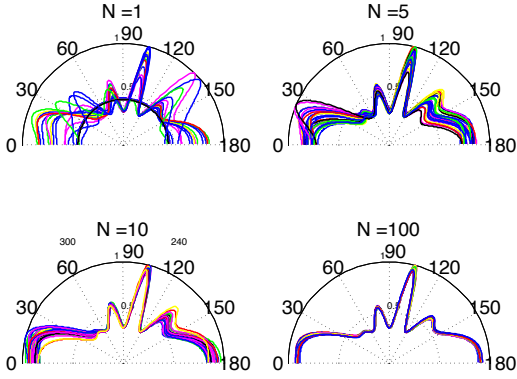


Figure 18: The effect of number of data samples on AoA spectrum.

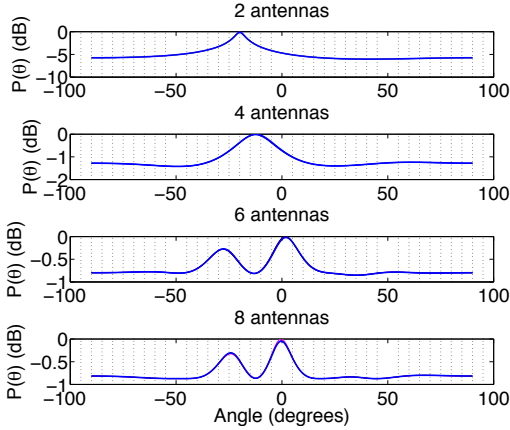


Figure 19: More antennas improve resolution and accuracy. Resolution and accuracy benefit localization.

For a 1000-byte packets transmitted at 6 Mbps, the transmission time is around $1000 * 8 / 6 = 1.33$ ms. We note that the total latency for Array Track is around 5 ms:

$$T_l = T_d + T_t + T_p - T = 5ms \quad (20)$$

This is a quite small latency and it can be further decreased if we improve the Ethernet connection speed between WARP and PC and process with a more powerful PC.

4.4 Effect of number of antennas

We show the effect of number of antennas on the AoA signatures in this section. In order to obtain accurate bearing information in indoor environment with strong multipath, enough number of antennas is necessary. When there are no multipath reflections and the direct path signal is relatively strong, even two and four antennas can generate quite accurate results. However, in a more challenging environment when the direct path is relatively weak and multipath reflections are strong, increasing the number of antennas will result in more

accurate bearing estimation. Another benefit of having more antennas is that the resolution of the pseudospectrum improves as the resolution can be achieved for N antennas is $180/N$ degrees which means if there are two signals coming within $180/N$ degrees range, these two signals can not be differentiated with N antennas. In Figure 19, we show the AoA pseudospectra plot for the same packet with two, four, six and eight antennas. We plot them in Cartesian coordinates as it's more clearly viewed than polar plot. A two-antenna arrangement generates one peak. Four antennas yield better resolution than two antennas with the measured bearing closer to the true bearing. However, with four antennas, it is not possible to differentiate two incoming signals within a 45 degree range. The direct path and reflected path are around 30 degrees apart from each other, so they can not be differentiated with four antennas. Instead, one peak at an angle between the two incoming signals is generated. However, this bearing is usually close to the true bearing. Once six antennas are applied, we find that both the direct path and multipath components are visible. With eight antennas, we have even better resolution and more accurate results. With future wireless access point designs (such as SAM [23] and MUBF [2]) scaling up the number of antennas at the access point, the trend favors our design.

4.5 Spatial smoothing

Traditional MUSIC algorithm is known to work poorly with highly correlated signals which the estimation accuracy is greatly degraded when the signals are coherent. As the direct path and reflection path signals in indoor environment are highly correlated, we implement MUSIC algorithm with spatial smoothing (SSP) and show the effect of spatial smoothing on real indoor multipath signals. As presented in Figure 20, we show the AoA spectra generated without SSP, with 2 groups of sub-array, 3 groups and 4 groups spatial smoothing. With more groups of sub-array applied, the effective number of antennas are decreased. 2 groups of sub-array have 7 effective antennas while 3 groups has 6 effective antennas. It can be shown that spatial smoothing can sharpen the peak for more accurate results. However, with more groups of sub-arrays, spatial smoothing has the negative effect of eliminating smaller peaks which is sometimes the direct path bearing peak. A good compromise is choosing 2 groups of sub-arrays which employs the advantage of spatial smoothing and not eliminating those small peaks.

5. RELATED WORK

ArrayTrack owes its research vision to early indoor location service systems that propose dedicated infrastructure to provide a fine-grained indoor location service. Active Badge [26] equips mobiles with infrared

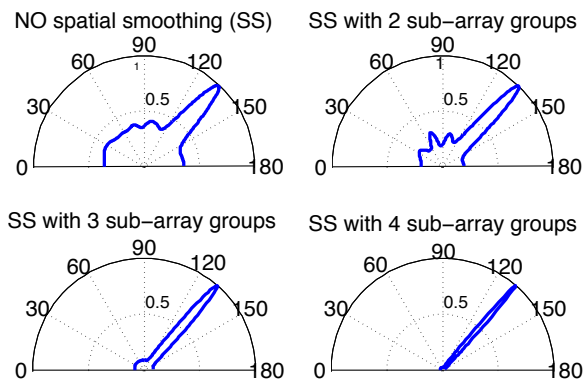


Figure 20: Spatial smoothing effects on AoA spectrum.

transmitters and buildings with many infrared receivers; active badges emit unique codes, which are then detected by the infrared sensor and associated with location with a six meter range. The Bat System [27] uses a matrix of RF-ultrasound receivers, each hard-coded with location, deployed on the ceiling indoors. Users wear “Bats” that transmit unique identifiers to the receivers over RF while sending simultaneous ultrasonic “chirps”. Cricket [16] equips buildings with combined RF-ultrasound beacons which transmit data with simultaneous ultrasound chirps, while mobiles carry RF and ultrasound receivers. Both Bat and Cricket measure time differences between the RF and ultrasound arrival, triangulating location by combining multiple measurements to or from different beacons.

The most widely used physical layer 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 physical-layer information, 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 [4, 5] builds “maps” of signal strength to one or more access points, achieving an accuracy on the order of meters [18, 22]. Later systems such as Horus [32] 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 converge area, but require large amounts of calibration. While some work has attempted to reduce the calibration overhead [10], mapping generally requires significant calibration effort. Other map-based work has proposed using overheard GSM signals from nearby towers [25], or dense deployments of desktop clients [3]. In contrast

to map-based techniques, the experimental results we show here achieve their accuracy with just one to two access points, and require no calibration beforehand.

Model-based approaches.

The second line of work using RSS are techniques based on mathematical models. Some of these proposals use RF propagation models [17] to predict distance away from an access point based on signal strength readings. By triangulating and extrapolating using signal strength models, TIX [9] achieves an accuracy of 5.4 meters indoors. Lim *et al.* [11] 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 [7] 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 two meters indoors, without the need for any explicit pre-deployment calibration.

Other model-based proposals augment RF propagation models with Bayesian probabilistic models to capture the relationships between different nodes in the network [12], and develop conditions for a set of nodes to be localizable [31]. Still other model-based proposals are targeted towards ad hoc mesh networks [6, 19, 15].

Prior work in AoA.

Wong *et al.* [28] 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.* [13] simulate AoA-based localization in an ad hoc mesh network. AoA has also been proposed in CDMA mobile cellular systems [30], in particular as a hybrid approach between TDoA and AoA [8, 29], and also in concert with interference cancellation and ToA [24].

Geo-fencing [21] utilizes directional antennas and a frame coding approach to control the indoor coverage boundary. Each AP sends out partial frames and only the clients located in the overlapping region covered by multiple APs can decode all the packets. Compared to Geo-fencing, ArrayTrack provides a location service and does not impact the arrangement of traffic in the wireless network.

Zhang *et al.* [14] propose a system that uses the channel impulse response and channel estimates of probe tones to detect when a device has moved.

6. CONCLUSION

We have presented ArrayTrack, an indoor location system that uses angle-of-arrival techniques to locate wireless clients indoors in a wireless local area network. ArrayTrack combines best of breed algorithms for AoA based direction estimation and spatial smoothing with novel algorithms for suppressing non-line of sight reflections that occur frequently indoors. ArrayTrack achieves 30 cm location accuracy when clients are stationary and 60 cm accuracy when moving at a walking speed, using just one access point equipped with eight antennas.

7. 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] E. Aryafar, N. Anand, T. Salonidis, and E. W. Knightly. Design and experimental evaluation of multi-user beamforming in wireless LANs. In *Proceedings of ACM MOBICOM*, 2010.
- [3] 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.
- [4] 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.
- [5] P. Bahl, V. Padmanabhan, and A. Balachandran. Enhancements to the RADAR location tracking system. Technical Report MSR-TR-2000-12, Microsoft Research, Feb. 2000.
- [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] K. Chintalapudi, A. Iyer, and V. Padmanabhan. Indoor localization without the pain. In *Proc. of ACM MobiCom*, 2010.
- [8] 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.
- [9] Y. Gwon and R. Jain. Error characteristics and calibration-free techniques for wireless LAN-based location estimation. In *ACM MobiWac*, 2004.
- [10] 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.
- [11] H. Lim, C. Kung, J. Hou, and H. Luo. Zero configuration robust indoor localization: Theory and experimentation. In *Proc. of IEEE Infocom*, 2006.
- [12] D. Madigan, E. Einahrawy, R. Martin, W. Ju, P. Krishnan, and A. Krishnakumar. Bayesian indoor positioning systems. In *Proc. of IEEE Infocom*, 2005.
- [13] D. Niculescu and B. Nath. Ad-hoc positioning system (APS) using AoA. In *Proc. of IEEE Infocom*, 2003.
- [14] N. Patwari and S. Kaser. Robust location distinction using temporal link signatures. In *Proc. of the ACM MobiCom Conf.*, pages 111–122, Sept. 2007.
- [15] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Mobile-assisted localization in wireless sensor networks. In *Proc. of IEEE Infocom*, 2005.
- [16] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *Proc. of the ACM MobiCom Conf.*, pages 32–43, Aug. 2000.
- [17] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice-Hall, 2nd edition, 2002.
- [18] T. Roos, P. Myllymaki, and H. Tirri. A probabilistic approach to WLAN user location estimation. *International J. of Wireless Information Networks*, 9(3), 2002.
- [19] A. Savvides, C. Han, and M. Srivastava. Fine-grained localization in ad-hoc networks of sensors. In *Proc. of ACM MobiCom*, 2001.
- [20] T.-J. Shan, M. Wax, and T. Kailath. On spatial smoothing for direction-of-arrival estimation of coherent signals. *IEEE Transactions on Acoustics, Speech and Signal Processing*, (4), 1985.
- [21] 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.
- [22] A. Smaligic, D. Siewiorek, J. Anhalt, D. Kogan, and Y. Wang. Location sensing and privacy in a context aware computing environment. In *Pervasive Computing*, 2001.
- [23] 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.
- [24] A. Tarighat, N. Khajehnouri, and A. Sayed. Improved wireless location accuracy using antenna arrays and interference cancellation. 4, 2003.
- [25] A. Varshavsky, E. Lara, J. Hightower, A. LaMarca, and V. Otsason. GSM indoor localization. In *Pervasive and Mobile Computing*, 2007.
- [26] 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.
- [27] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, 1997.
- [28] 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.
- [29] Y. Xie, Y. Wang, P. Zhu, and X. You. Grid-search-based hybrid ToA/AoA location techniques for NLOS environments. *IEEE Comms. Letters*, 13(4):254–256, 2009.
- [30] L. Xiong. A selective model to suppress nlos signals in angle-of-arrival (AoA) location estimation. In *Proc. of the IEEE PIMRC*, 1998.
- [31] Z. Yang, Y. Liu, and X. Li. Beyond trilateration: On the localizability of wireless ad-hoc networks. In *Proc. of IEEE Infocom*, 2009.
- [32] M. Youssef and A. Agrawala. The Horus WLAN location determination system. In *Proc. of ACM MobiSys*, 2005.