ArrayTrack: A Fine-Grained Indoor Location System

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

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

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 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. In order to provide indoor localization service, we need to differentiate the direct transmission path and reflection paths as the direct path contains the location information. One method is to identify which signal arrives first as the direct path is always the shortest. As in indoor environment, the length difference between direct path and reflection paths is pretty small, corresponding to a small time difference (about 20-50 ns), it's challenging to differentiate them as it requires expensive hardware to sample the signals at very high rate. As the existing 802.11 a/b/g occupy 20 MHz bandwidth, most commercial hardwares employ 40 MHz sampling rate. This rate is far below that required to differentiate indoor multipath signals in time domain.

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.

We propose a system ArrayTrack that offers a centimeter-accurate location service which utilize the widely-deployed WiFi access points (AP) without any dedicated infrastructures needed. The key observation we make 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 particular, exploits MIMO extensively through the use of many antennas at the access point. A lot of commercial APs from different manufacturers with multiple antennas are already on market as shown in Figure 1. 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 [13, 1].

Figure 1: Commercial Access Points (AP)

2. RELATED WORK

ArrayTrack owes its research vision to early indoor location service systems that propose dedicated infrastructure to provide a indoor location service including active Badge [14], the Bat System [15] and Cricket [8]. All these systems requires dedicated infrastructures to be installed for location purpose while ArrayTrack utilizes existing WiFi APs widely deployed.

The most widely used physical layer information for localization is received signal strength (RSS). While readily available from commodity WiFi hardware, the resulting RSS measurements are very coarse, especially when few readings are present. 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 [10, 12]. Later systems such as Horus [17] use probabilistic techniques to improve localization accuracy to an average of 0.6 meters with an average of six access points and large amounts of calibration. In contrast, ArrayTrack achieves better accuracy with just one to two access points, and require no calibration beforehand.

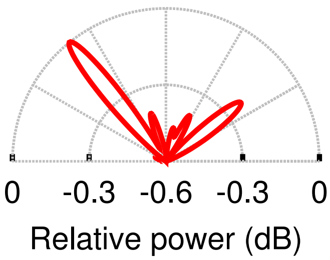

Figure 2: A *pseudospectrum* of a client's received signal at a multi-antenna access point estimates the incoming signal's power as a function of its angle of arrival.

The second line of work using RSS are techniques based on mathematical models. Some of these proposals use RF propagation models [9] to predict distance away from an access point based on signal strength readings. By triangulating and extrapolating using signal strength models, TIX [5] achieves an accuracy of 5.4 meters indoors while Lim *et al.* [6] achieves a localization error of about three meters indoors. Our ArrayTrack system is much simpler and achieves better accuracy.

Wong et al.[16] investigate the use of AoA and channel impulse response measurements for localization. Their scheme requires a signal with a very high SNR (60 dB) which is not possible in real world and the authors stop short of describing a complete system design.

Zhang *et al.* [7] propose a system that uses the channel impulse response and channel estimates of probe tones to detect when a device has moved. However, this system is not able to obtain the location information but only whether the client was moved which is very limited information.

3. DESIGN AND CONTRIBUTIONS

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). The result, shown in Figure 2, 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 maximum peak does not always correspond to the direct path. To address this problem, we introduce a novel architecture and multipath disambiguation algorithm to differentiate the direct path and reflection paths.

Another 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 [4].

Another uniqueness of ArrayTrack is its robustness against low SNR and colliding. ArrayTrack works well with very low SNR as it does not decode any part of the packet. It only detects the most robust part of the preamble part of a packet. Furthermore, by applying the proposed creative iterative method, we are able to obtain bearing information for two collided packets.

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 clients we tested, ArrayTrack can consistently localize clients 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.

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

3.1 Packet detection

To obtain the bearing information for a client, the AP needs to overhear some transmissions from this client. ArrayTrack only needs a very tiny part of the packet to process the angle of arrival (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 furthermore, 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. We implemented modified version of Schmidl-Cox algorithm on WARP FPGA to detect the preamble of a packet. Once a packet is detected, multiple samples of the packet are recorded to process AoA spectrum.

3.2 AoA spectrum generation

We first describe how to compute angle of arrival when there is just one signal 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 3(a).

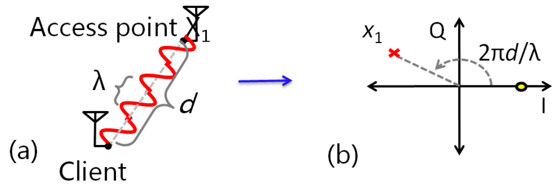

Figure 3: ArrayTrack's principle of operation: (a) The phase of the signal goes through a 2π cycle every radio wavelength λ. (b) The complex representation of the sent (filled dot) and received (crosses) signals at the antenna in (a).

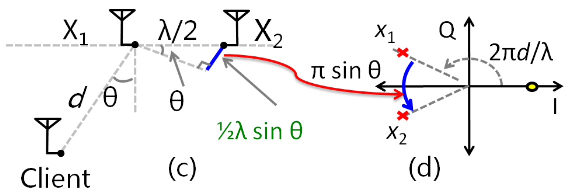

Figure 4: (c) A signal arriving at bearing θ to two antennas. (d) The complex representation of the sent (filled dot) and received (crosses) signals at both antennas in (c).

This means that the access point receives signals with an added phase determined by the path length d from the client. Phase is particularly easy to analyze as the phase of the wireless signal is graphically represented using an *in-phase-quadrature* (I-Q) plot, as shown in Figure 3(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 $2\pi d/\lambda$. 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 4(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. Assume the distance d is much larger than $\lambda/2$ and apply simple mathematics, this amount of extra distance is calculated as:

$$\Delta d = \lambda/2 * \sin\theta \tag{1}$$

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 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 θas:

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

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. Adding antennas can resolve the ambiguity. The best known AoA estimation algorithms are MUSIC [11] algorithm based on eigenstructure analysis.

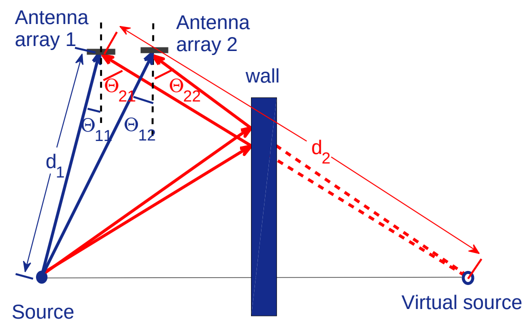

Figure 5: Identify direct path bearing by distance comparison: $d_1 < d_2 \Rightarrow \theta_{11}$ is direct path bearing.

3.3 Direct and reflection paths differentiation

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, it's critical for us to identify the direct path bearing for localization. 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. 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. We propose a novel scheme here 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 5. Two antenna arrays are placed close to each other in one line. 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 5 arrives at the two arrays with another two close bearings. By knowing these two pairs of bearings, we are able to obtain d_1 and d_2 indicated in Figure 5 receptively:

$$d_1 = \left|\frac{1}{\tan\theta_{11} - \tan\theta_{12}} * \frac{1}{\cos\theta_{11}}\right| \tag{3}$$

$$d_2 = \left|\frac{1}{\tan\theta_{21} - \tan\theta_{22}} * \frac{1}{\cos\theta_{21}}\right| \tag{4}$$

As we know, the direct path distance is always the shortest. d_1 and d_2 are compared to identify the direct

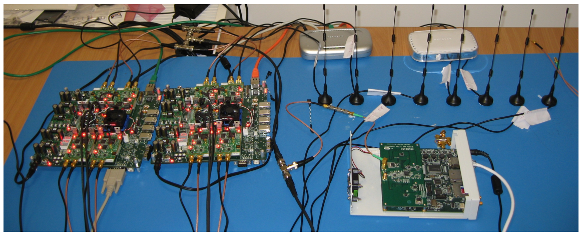

Figure 6: The ArrayTrack prototype with linear antenna arrangement.

path bearing. If there are more than two pairs, the minimum distance pair is chosen.

4. RESULTS AND EVALUATIONS

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.

Our prototype AP as shown in Figure 6 uses two Rice WARP FPGA-based wireless platforms. Each WARP platform is equipped with four omindirectional antennas. All the eight antennas are phase calibrated. An ordinary laptop with WiFi serves perfectly as a client working with our ArrayTrack implemented AP. We use the Soekris boxes as our clients as they are small and cheap. We place the prototype access point at the point marked "AP" in our testbed floorplan, shown in Figure 7. 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 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 and show the results in Figure 7.

The x axis represents the true bearing we measure from the map. The y axis is the bearing we obtain from ArrayTrack. If the mean value is close to the blue line, it means the bearing obtained is accurate. By applying our differentiation scheme, the results are presented in

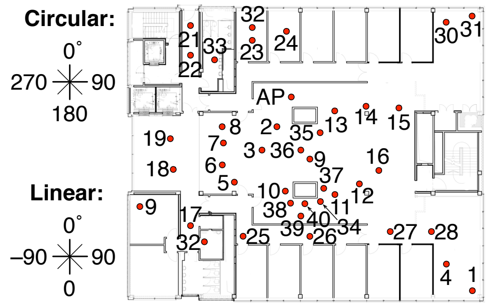

Figure 7: Testbed environment: Soekris clients are numbered, and the WARP "AP' is labeled.

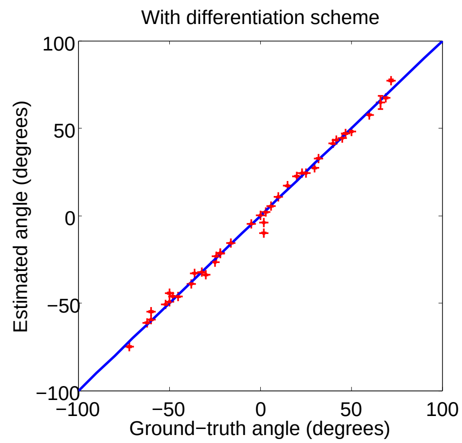

Figure 8: Measured versus ground truth bearing estimation for Soekris clients in the office environment with path differentiation scheme.

Figure 8: only one client's bearing is not detected accurately; the bearings of all other 39 clients are estimated very accurately with a mean error of 1.3 degree regardless of proximity to the AP, location inside or outside the AP's room. The 99% confidences show the stableness of our scheme with time.

There is one client (client 39 in Figure 7) which our scheme fails. There are two pillars in-between client 39 and the AP. ArrayTrack is able to differentiate the direct path and reflection path efficiently. However, in some extreme case when the direct path is 100% blocked by metal or in our scenario here by two pillars, ArrayTrack may fail. 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 except in

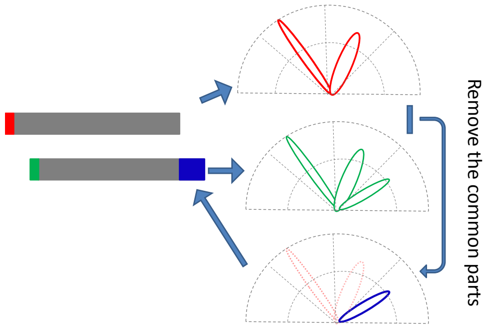

Figure 9: The procedure to obtain AoA spectra for two colliding packets.

some extreme scenarios when the direct path is totally blocked and this is very rare in real indoor environment.

4.2 Robustness

Robustness is one important characteristic ArrayTrack wants to achieve. As ArrayTrack works with the preamble part of a packet, ArrayTrack is robust against low SNR. 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. 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 size 1000 bytes, if collision happens, the percentage of these training symbols (assume we utilize all the 10 short training symbols (6 bytes) to detect the packet) get collided is 0.6% which is pretty small.

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 9. 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. 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.

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 10. 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

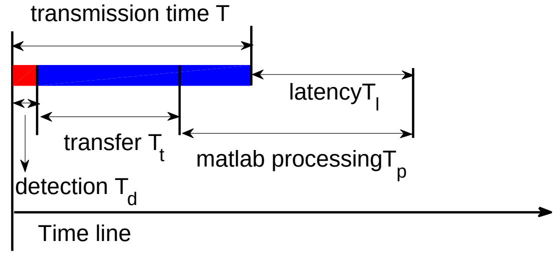

Figure 10: The latency ArrayTrack system introduces.

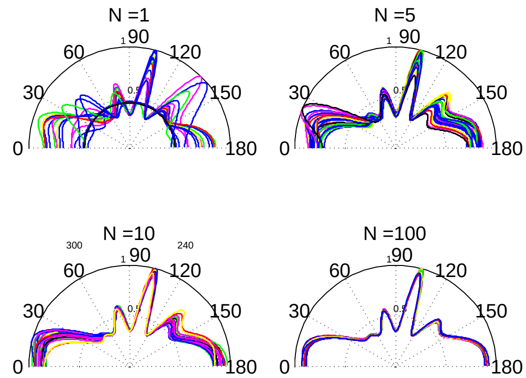

Figure 11: The effect of number of data samples on AoA spectrum.

T_t: Assume N samples are to be transmitted from WARP to PC for processing and speed is S Mbps. Each sample is 4 bytes and we have 8 radio boards. So the time needed for the transmission is calculated as: T_t=N * 32 * 8/(1000 * S) ms. Due to the simple IP stack currently implemented on WARP, the maximum throughput can be achieve is 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 11. We can see clearly that when the number of samples increased to 5, the AoA spectrum is already quite stable.

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

T_t = 5 * 32 * 8 /1000 =1.28 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. This time has the potential to be decreased significantly in the future.

For a 1000-byte packet transmitted at 6 Mbps, the transmission time is around 1000 * 8 /6 = 1.33 ms. So the current latency ArrayTrack introduces ($T_d + T_t + T_p - T$) is around 5 ms and it can be further decreased if we improve the Ethernet connection speed between WARP and PC and process with a more powerful PC. It's very likely for us to achieve 0 latency which means the whole processing is finished before the packet is 100% transmitted in the future.

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

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

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

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

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

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

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

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

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