

General-Purpose Deep Tracking Platform across Protocols for the Internet of Things

Zhenlin An*, Qiongzheng Lin*, Ping Li[†], Lei Yang**

*Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR

[†]School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China

an@tagsys.org, lin@tagsys.org, pingli0112@gmail.com, young@tagsys.org

ABSTRACT

In recent years, considerable effort has been recently exerted to explore the high-precision RF-tracking systems indoors to satisfy various real-world demands. However, such systems are tailored for a particular type of device (e.g., RFID, WSN or Wi-Fi). With the rapid development of the Internet of Things (IoT), various new wireless protocols (e.g., LoRa, Sigfox, and NB-IoT) have been proposed to accommodate different demands. The coexistence of multiple types of IoT devices forces users to deploy multiple tracking systems in a warehouse or a smart home where various IoT devices are running, which causes huge additional costs in installation and maintenance. To address this issue, this work presents iArk, which is a general-purpose tracking platform for all types of IoT devices working at the ultra high frequency band. Our innovation lies in the design of the “K+1”-model hardware, the protocol free middleware, and the multipath resistant learnware. By the virtue of decoupling from wireless protocols, iArk also allows researchers to concentrate on developing a new tracking algorithm without considering the protocol diversity. To date, the platform can support five mainstream types of IoT devices (i.e., NB-IoT, LoRa, RFID, Sigfox and Zigbee) and is scalable to other types with minimal effort.

CCS CONCEPTS

- Human-centered computing → Ubiquitous and mobile computing;
- Computing methodologies → Artificial intelligence.

KEYWORDS

Internet of Thing, Localization, Deep learning

ACM Reference Format:

Zhenlin An*, Qiongzheng Lin*, Ping Li[†], Lei Yang*. 2020. General-Purpose Deep Tracking Platform across Protocols for the Internet of Things. In *The 18th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '20), June 15–19, 2020, Toronto, ON, Canada*. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3386901.3389029>

*Zhenlin An and Qiongzheng Lin are the co-primary authors. Lei Yang is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MobiSys '20, June 15–19, 2020, Toronto, ON, Canada

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7954-0/20/06...\$15.00

<https://doi.org/10.1145/3386901.3389029>



Figure 1: Smart Warehouse Equipped with Different Types of IoT Device.

1 INTRODUCTION

The Internet of Things (IoT) extends the Internet connectivity to physical devices and everyday objects. IoT affects our daily lives in many domains, ranging from tiny wearable devices to large industrial systems. High-precision tracking has prompted a series of key IoT applications. Especially, tracking IoT devices inside buildings where GPS is unavailable has become a growing business interest.

Meanwhile, many IoT protocols have been rapidly developed in recent years to meet a variety of demands such as RFID, ZigBee, LoRa, Sigfox, and NB-IoT. Fig. 1 shows how various IoT devices create smart warehouses¹. Their target application scenarios are clearly diverse. NB-IoT and LoRa nodes are installed to track transporters outdoors and forklifts indoors, respectively; RFID tags are attached to goods for auto-identification and management; Sigfox tags are used to monitor raw materials and dangerous chemicals; temperature or light sensors over Zigbee are deployed to measure the warehouse environment. Such diversity in wireless protocols, however, has become a serious challenge in device management and tracking. At present, the academia and the industry usually use the divide-and-conquer paradigm to handle diversity. Existing solutions have been tailored for specific types of devices. For example, several works [31, 33, 46, 48, 49] targeted the localization of RFID tags; one work [16] was designed for Zigbee mote positioning, and another [27] concentrated on LoRa nodes. An all-in-one platform for tracking all types of IoT devices does not exist. Thus, users must deploy multiple tracking systems in a warehouse or a smart home where various IoT devices are running or equip a single device with multi-radios. We predict that the coexistence of multiple protocols in IoT is more likely to occur than the outright success of any single one in the near future for two main reasons.

¹A practical smart warehouse may only adopt some of IoT protocols. This toy example shows an extreme case.

Table 1: Summary of Mainstream IoT Technologies

Type	Freq. (MHz)	BW	Date rate	Modulation
RFID	902 - 928	500 kHz	40 kbps	ASK/PSK
NB-IoT	880 - 915	180 kHz	15 kbps	BPSK/QPSK
Zigbee	868/915	1 MHz	250 kbps	BPSK/OQPSK
LoRa	868/915	500 kHz	50 kbps	CSS
Sigfox	868/915	200 Hz	600 bps	DBPSK/GFSK

*The above IoT technologies can work at multiple bands (e.g., 2.4 GHz). This table only shows their technical parameters at UHF band (i.e., 860 ~ 960 MHz) in accord with the North American standards.

First, no single technology can fully meet the complex application requirements. For example, RFIDs are low cost but have a limited range (a few meters). Second, replacing legacy devices completely is time consuming even though technology evolves rapidly. Therefore, a single technology is unlikely to dominate the current IoT market. Consequently, the difficulties in management and tracking due to the coexistence of multiple protocols will persist in the industry for a long time. In particular, the long-term maintenance cost for the warehouse management system will persistently introduce much overhead and cause a huge rise in the manufacturing cost, make the manufacturer lost their competitiveness.

In this work, we present iArk, which is the first general-purpose and all-in-one tracking platform for the *non-intrusive*, *high-precise* and *flexible* localization of different types of IoT devices in the presence of severe multipath effects. This triangulation-based tracking platform is powered by a large-sized antenna array with two merits. First, this solution is non-intrusive. That is, the platform acts as a protocol-free sniffer simply reporting location results with a timestamp over the phase measurement. It does not need to communicate with IoT devices. Thus, the platform itself is essentially protocol-free without knowing the specific models. Second, the antenna array can separate the line-of-sight (LOS) direction in a complex setting full of multipath propagations to perform accurately. Table 1 summarizes the technical parameters of the mainstream IoT technologies, which prefer the ultra-high frequency (UHF) ISM band as one of operating bands for improved penetrability indoors (especially in an industrial setting) and a license-free spectrum. This condition provides an opportunity to build a platform that is compatible with multiple types of IoT devices at the UHF band.

However, constructing such a tracking platform requires addressing the following challenges of engineering hardware, middleware, and learnware:

Hardware: Achieving a *high-precision* tracking platform requires considerable effort in hardware design. Specifically, building an RF frontend composed of large-sized antenna array is nontrivial. Scaling the baseband processing, transmission synchronization, reducing onboard losses and cost control raise serious system challenges. Thus, only testbeds with few antennas for indoor localization have been reported, such as 16 antennas in ArrayTrack [52] and 12 antennas in SWAN [51]. Argos [45] is a pioneer work in building large-scale antenna array but does not work for indoor localization. In this work, we build an iArk prototype with an 8 × 8-element antenna array plus a side antenna, by using commercial off-the-shelf radio modules, as shown in Fig. 2. Notice that the RF frontend is the most expensive component in the communication system. Given the cost-effectiveness of the prototype, we leverage 9 RF switches

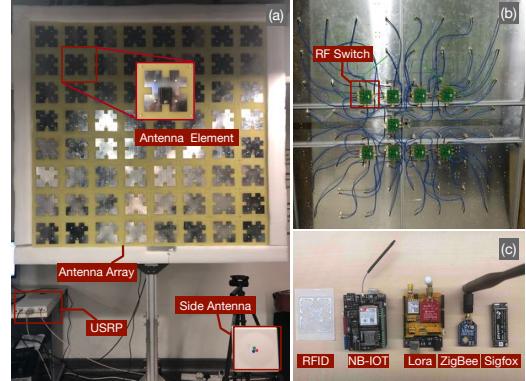


Figure 2: Deep Tracking Platform. (a) and (b) show the frontside and backside of the antenna array, respectively; (c) shows the tested IoT devices.

to string the 64 array elements, that share a single *main-channel* (i.e., connected to a single RF frontend) in a time-slicing manner. One of our innovations is the design of the *side-channel*, which shows considerable merit in dealing with the protocol diversity. Additional details are presented in §4.

Middleware: Previous indoor localization systems are usually bound to a specific type of device because the existing signal-processing algorithms for estimating physical-layer parameters (e.g., RSS or phase) are protocol-specific. For example, the computation of the angle of arrival (AoA) only considers the carrier phase rotation over the distance. However, the phase is also affected by other factors, such as preamble structure, modulation and encoding schemes, and operating channel. As a result, the cross-protocol phase estimation is a daunting undertaking. To address this challenge, we present a novel *protocol-free estimation algorithm* that leverages the signals from the side channel to counteract the negatives from the main channel. In §5, we explore the preceding idea further and describe the manner by which the algorithm can work with the dual signals from the frontend.

Learnware: For a general-purpose platform across protocols, past tracking approaches can be easily migrated to iArk with minimal efforts. To demonstrate this capability, we develop an AI-boosted localization system. Specifically, we logically divide the 8 × 8 array into 25 overlapping 4 × 4 subarrays as inspired by the binocular stereo in computer vision; each subarray can estimate a direct AoA of the device. The targeted device will be exactly located at the intersection of the lines along the estimated angles. The system works well in a relatively stationary environment in the absence of multipath. However, a warehouse is full of indoor reflectors (e.g., ceiling, walls, and furniture) in reality and these reflectors produce many multipath propagations. Consequently, the estimated AoA is severely skewed. To address this challenge, we use two deep neural networks, the *AoA neural network* (ANN) and the *Triangulation neural network* (TNN), which are used to compute the AoA and perform the triangulation respectively in §6.

Summary of Results. We evaluate five types of off-the-shelf IoT devices. The results demonstrate that iArk can fully identify the direct AoA with median errors of (0.46°, 1.5°) at azimuthal and elevation angles. It performs localization with median errors of 8.7,

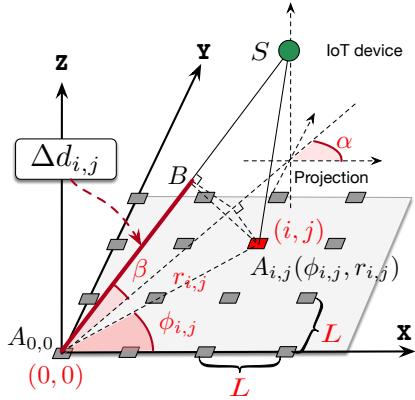


Figure 3: AoA Computation with an Antenna Array. K antennas (aka elements) are deployed in a grid. The distance difference $\Delta d_{i,j} = |SA_{i,j}| - |SA_{0,0}| = -r_{i,j} \cos(\alpha - \phi_{i,j}) \cos(\beta)$.

5.8 and 14.2 cm in X, Y and Z dimensions over a medium-sized office ($10 \times 20 \text{ m}^2$). The accuracy of iArk matches or even exceeds that of past AoA systems.

2 PRELIMINARY

Before introducing our system, a brief review is given in terms of the antenna array and spatial spectrum.

Antenna Array. The direction of the RF source can be uniquely represented with two angles, azimuthal angle and elevation angle. The Angle of Arrival (AoA) of the RF source is computed by comparing the phases of the received signals at multiple antennas. Suppose an antenna array with $\sqrt{K} \times \sqrt{K}$ antennas (aka elements). The spacing of two adjacent antennas is L , which is less than the wavelength λ , i.e., $L < \lambda$. Fig. 3 shows an example of 4×4 antenna array. For the convenience of calculation, the positions of antennas are transformed to the polar coordinate system where the antenna $A_{0,0}$ on the bottom-left corner is selected as the origin. The polar coordinate of the antenna $A_{i,j}$ is given by:

$$\begin{cases} r_{i,j} = L\sqrt{i^2 + j^2} \\ \phi_{i,j} = \arctan 2(j, i) \end{cases} \quad (1)$$

where $i, j = 0, \dots, \sqrt{K} - 1$. Let $d_{i,j}$ be the distance from the RF source S to the antenna $A_{i,j}$. When S is relatively far from the array, $|SA_{i,j}| \approx |SB|$ where $A_{i,j}B \perp SA_{0,0}$. The distance difference between the source S and the antennas of $A_{i,j}$ and $A_{0,0}$, denoted by $\Delta d_{i,j}$, is given by:

$$\Delta d_{i,j} = d_{i,j} - d_{0,0} = -r_{i,j} \cos(\alpha - \phi_{i,j}) \cos(\beta) \quad (2)$$

where α and β denote the azimuthal and elevation angles, respectively, and $\alpha \in [0, 360^\circ]$ and $\beta \in [0, 90^\circ]$. Let $\theta_{i,j}$ denote the phase of the RF signal measured at the antenna $A_{i,j}$, $\theta_{i,j} \in [0, 2\pi]$. The phase and the distance have the following known relation at antenna $A_{i,j}$ due to the phase rotation:

$$\theta_{i,j} = 2\pi d_{i,j}/\lambda \bmod 2\pi \quad (3)$$

Hence, the phase difference between the received signals at the two antennas, denoted by $\Delta\theta_{i,j}$, relates to the difference in their

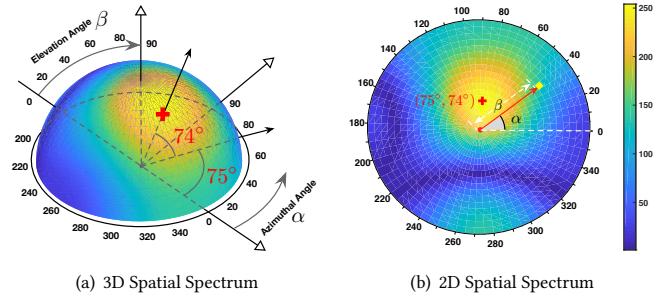


Figure 4: Illustration of a Spatial Spectrum. The spatial spectrum is generated by a 2×2 antenna array. (a) and (b) show the same spatial spectrums but in 3D and 2D forms respectively. The 2D spatial spectrum is the projection of the 3D spectrum.

distances from the source as follows:

$$\begin{aligned} \Delta\theta_{i,j} &= \theta_{i,j} - \theta_{0,0} = 2\pi\Delta d_{i,j}/\lambda \bmod 2\pi \\ &= -2\pi r_{i,j} \cos(\alpha - \phi_{i,j}) \cos(\beta)/\lambda \bmod 2\pi \end{aligned} \quad (4)$$

Spatial Spectrum. Copies of a signal traveling along different paths may overlap at the receiving antenna. To separate the copies of the device’s signal arriving from different directions, we must measure the power coming from each direction. A large phased antenna array can achieve this goal by forming a very narrow beam and steering it around. When steering its beam to the line-of-sight (LOS) direction, the array essentially filters out the power coming from all other directions. Equivalently, we can compute a *spatial spectrum* (SS) [33, 49, 54] that indicates all possible AoAs. Formally, SS is defined as $P(\alpha, \beta)$, where the power of the signal received in the beam from azimuthal angle α and elevation angle β . Let $\theta_{i,j}$ and $\widehat{\theta}_{i,j}$ denote the theoretical and measured phases of the RF signal received at the antenna $A_{i,j}$. We can derive the relative power of the source along the angle of (α, β) as follows:

$$\begin{aligned} P(\alpha, \beta) &= \left| \frac{1}{K} \sum_{i,j}^{\sqrt{K}, \sqrt{K}} e^{j(\Delta\widehat{\theta}_{i,j} - \Delta\theta_{i,j})} \right|^2 \\ &= \left| \frac{1}{K} \sum_{i,j}^{\sqrt{K}, \sqrt{K}} e^{j((\widehat{\theta}_{i,j} - \widehat{\theta}_{0,0}) - \frac{-2\pi r_{i,j}}{\lambda} \cos(\alpha - \phi_{i,j}) \cos(\beta))} \right|^2 \\ &= \left| \frac{1}{K} \sum_{i,j}^{\sqrt{K}, \sqrt{K}} e^{j(\widehat{\theta}_{i,j} + \frac{2\pi r_{i,j}}{\lambda} \cos(\alpha - \phi_{i,j}) \cos(\beta))} \right|^2 \end{aligned} \quad (5)$$

where $\Delta\widehat{\theta}_{i,j} = \widehat{\theta}_{i,j} - \widehat{\theta}_{0,0}$ and $\Delta\theta_{i,j}$ is defined in Eqn 4. $\widehat{\theta}_{0,0}$ is a constant term that can be extracted to outside of the sum, and $|e^{-j\widehat{\theta}_{0,0}}| = 1$. The aforementioned equation actually correlates the measured phase difference with a “template” across the K antennas given an AoA of (α, β) . The SS would spike at the LOS direction of the RF source. Thus, the direction leading to the maximum power in the SS is the direct AoA (i.e., LOS direction). To visually understand the SS, we show an example in Fig. 4. The SS is generated by a 2×2 antenna array. The maximum relative power of SS is achieved at the angles of $(75^\circ, 74^\circ)$, which is considered as the direct AoA of the RF source. The polar 2D SS is a projection of the 3D SS, where the polar angle and distance in the 2D SS indicate the azimuthal

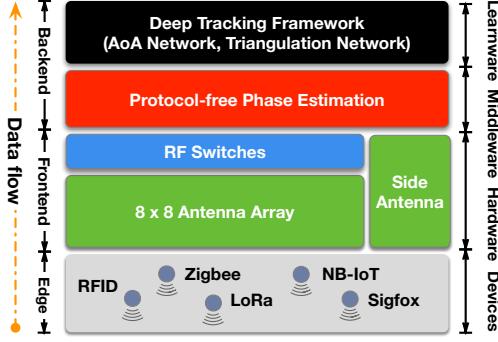


Figure 5: System Architecture

and elevation angles respectively. For the sake of clarity, we will use 2D SS in the subsequent sections by default.

Triangulation. After two direct AoAs at two different positions are obtained, an RF source can be located at the intersection of the two directions. This approach is called *triangulation*.

3 OVERVIEW

We build a novel platform called iArk, which consists of a large-sized two-dimensional antenna array, to measure the AoA for the triangulation. The platform is for all types of IoT devices operating at the UHF ISM band (i.e., approximately 915 MHz). It can be also easily extended to other frequency bands (e.g., 2.4 GHz) by simply replacing the RF frontend. iArk adopts a layered and loose-coupled architecture, as shown in Fig. 5. From a high-level perspective, the system contains three layers, namely, the hardware, the middleware, and the learnware.

- **Hardware:** We design an “8×8+1” frontend to acquire RF signals from the IoT devices. The “8×8” - antenna array can significantly increase the spatial diversity in the AoA estimation (§4).
- **Middleware:** We design a middleware layer to hide the tracking algorithm from the heterogeneity of wireless protocols. In particular, the protocol-free phase estimation (PPE) is constructed for this purpose (§5).
- **Learnware:** We develop AI-boosted localization framework called *learnware* over the middleware for case study. The learnware is composed of two deep neural networks for AoA estimation and triangulation respectively (§6).

The platform acts as a sniffer, so it does not need to know what kinds of devices in the warehouse and not to communicate with them. The platform also does not need to know which device transmits the signal or whether a signal collision occurs. This task is left to the upper-layer applications. The platform simply stores all location results over the phase matrices in the database with a timestamp. The transmitting device or any third-party (e.g., gateway) that is interested in the localization results should know when the device successfully transmits a packet. They can retrieve the locations through the timestamps. Thus, the platform can support any number of devices as long as they could successfully communicate with each other. Our design is essentially scalable for any kind of IoT network with a large number of devices. Such a loose-coupling architecture design avoids packet recognition, which benefits fast deployment and provides a measure for protecting privacy.

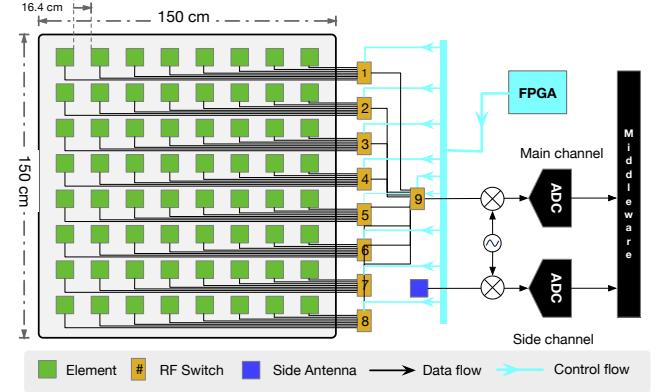


Figure 6: Schematic of the RF Frontend. The frontend comprises an 8×8 antenna array and a side antenna that form the main and side channels, respectively.

4 HARDWARE: ACQUIRING SIGNALS IN A NON-INTRUSIVE MANNER

At the heart of the hardware is the RF frontend, which is used to acquire RF signals from the air. Fig. 6 shows the schematic of our frontend. From a high level perspective, the frontend consists of an antenna array and a side channel. The frontend continues to listen at the 915 MHz wireless carrier with a 32.8 cm wavelength.

4.1 Switched Antenna Array

One of the key components of the frontend is an 8×8 antenna array. A total of 64 elements are deployed in the form of a grid with a spacing of 16.4 cm. Each array element consists of a directional square pitch with a side length of 12 cm (i.e., half wavelength). The array is 150 × 150 cm² in area. The ADC, which is an expensive component in the RF communication system, is used to convert an analog signal into a digital signal. The high-end antenna array used in ArrayTrack [52] and Argos [45] equips each element with a stand-alone ADC, which increases the entire cost of the antenna array by orders of magnitude. A few bits of the incoming packet are sufficient for each element to estimate the signal phase. Acquiring the entire packet is unnecessary for each element. To save cost, we allow all elements to share a single ADC by using 9 single-pole-eight-throws (SP8T) RF switches. Specifically, these switches constitute a two-stage switching system. The first stage is composed of eight switches (switches #1 ~ #8), each of which is connected to eight array elements in a row. In the second stage, the ninth switch (switch #9) further bridges the previous eight RF switches to an ADC. This design can fully utilize the 64 ports of the switches. These switches are manipulated by an FPGA to connect one of the 64 elements to the ADC instantly (with a delay of approximately 3μs). At any moment, only a single antenna element can be put through. In this manner, each element can receive a segment of the entire packet. The switching delay is negligible compared with the ms-level packet duration.

The 64 elements are scheduled to connect to the ADC in sequence. The scheduling cycle starts from the first element on the top-left corner and then goes through the others from left to right, top to

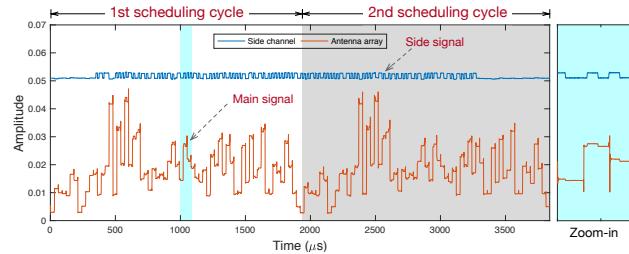


Figure 7: RF Signals Acquired via the Main and Side Channels. The signal is transmitted from an RFID tag. The red curve shows the main signal, which merges the segments acquired by the elements. Side channel continuously acquires the RF signal as shown in the blue curve.

bottom, and finally ends at the 64th element. The scheduling cycle is repeated. In each cycle, every element is exclusively connected to the ADC over 30 μ s. In total, each scheduling cycle takes (30 + 3) × 64 = 2112 μ s. Regarding the 6 MS/s sampling, we can obtain 30 μ s × 6MS/s = 180 samples from each element during a scheduled cycle. The 64 array elements and the ADC form the *main channel*.

4.2 Side-Channel

Unlike prior works [13, 23, 25, 37, 42, 51, 52], our unique design is the *side channel*, which is a second channel parallel to the main channel. The side channel is composed of an additional side antenna and a standalone ADC. It acquires RF signals persistently and continuously. Fig. 7 compares the signals acquired from the main and side channels. The signal contains many discontinuous “stairs”, each of which represents the signal acquired via one array element. The elements on the opposite corners are up to 163 cm apart and span over nearly 5 propagation cycles, which makes the signal segments differ in amplitude and phase. On the contrary, the signal through the side channel is considerably smoother and more continuous because it is acquired by the same antenna uninterruptedly. The merits of the side channel are discussed in §5.

5 MIDDLEWARE: GENERATING PROTOCOL-FREE SPATIAL SPECTRUM

The fundamental of AoA computation is the phase estimation of the RF signal as mentioned in §2. However, estimation is non-trivial, especially when handling numerous IoT devices that communicate with different protocols. In this section, we first investigate the current estimation algorithms to motivate our design and then describe our estimation algorithm.

5.1 Background of Phase Estimation

A wireless signal is typically represented by a stream of discrete complex numbers [34]. The digital bits cannot be directly propagated into the air. The transmitter should initially convert the digital bits into complex symbols. This step is called shift-keying. For example, Zigbee and NB-IoT map a “0” bit to $e^{j\pi}$ and “1” bit to $e^{-j\pi}$ (i.e., BPSK), and LoRa maps two bits into different chirp signals (i.e., chirp spectrum spread (CSS)). For simplicity, we use $X[t]$ to denote the transmitted symbols after shift-keying. Then,

the transmitter moves these symbols onto the carrier at frequency f_c by multiplying the symbols and the carrier signal, in a process called modulation. The transmitting carrier can also be represented by complex numbers, that is, $C_T[t] = e^{j(2\pi f_c t + \Phi)}$, where Φ is a constant (i.e., the initial phase shift caused by the hardware). Finally, the transmitted signal (i.e., $S_T[t]$) propagated into the air can be modeled as:

$$S_T[t] = X[t]C_T[t] = X[t]e^{j(2\pi f_c t + \Phi)} \quad (6)$$

At the receiver side, the received signal differs from the transmitted signal due to propagation. Specifically, the received signal (denoted by $S_R[t]$) turns into:

$$S_R[t] = H S_T[t] = a X[t] e^{j(2\pi f_c t + \theta + \Phi)} \quad (7)$$

where $H = ae^{j\theta}$ is the channel parameter. Amplitude a refers to the channel attenuation and its angle θ is a phase shift that depends on the distance between the transmitter and the receiver. To decode the transmitted symbols, the received signal is multiplied by the conjugate of a similar carrier signal (i.e., $C_R[t] = e^{j(2\pi f'_c t + \Delta\phi)}$) in a process called demodulation. The received symbol (i.e., $Y[t]$) is given by:

$$Y[t] = S_R[t] C_R^*[t] = a X[t] e^{j(2\pi \Delta f \cdot t + \theta + \Phi - \Delta\phi)} \quad (8)$$

where $\Delta f = (f_c - f'_c)$ is called the carrier frequency offset (CFO) and $\Delta\phi$ is the carrier phase offset (CPO) introduced by the up- and down-conversions. Essentially, CFO and CPO are results of the out-of-sync in frequency and clock between the transmitter and the receiver. The received symbol $Y[t]$ differs from the transmitted symbol $X[t]$ in amplitude and phase. If the transmitter and receiver are well-synchronized in frequency and clock, then $\Delta f = 0$ and $\Delta\phi = 0$. The received symbol can be further simplified as follow:

$$Y[t] = a X[t] e^{j(\theta + \Phi)} \quad (9)$$

Let $\tilde{\theta} = \angle \frac{Y[t]}{X[t]}$ where $\angle(\cdot)$ takes the angle of the complex number. Then,

$$\tilde{\theta} = \angle \frac{Y[t]}{X[t]} = \theta + \Phi \quad (10)$$

θ is the *true phase* that rotates over the distance. We call $\tilde{\theta}$ *pseudo-phase*, which is the sum of the true phase and the constant phase shift, for distinction. From the equation, we can estimate the pseudo-phase of the RF signal by comparing the transmitted with the received symbols. We call this approach *direct phase estimation* (DPE), which is widely adopted in existing systems.

5.2 Motivations and Challenges

Unfortunately, DPE fails to work for a large number of protocol-diverse IoT devices for four reasons:

- **Preamble dependence:** Eqn. 10 suggests that the pseudo-phase is the angle of the ratio of received symbols (i.e., $Y[t]$) to the transmitted symbols (i.e., $X[t]$). Thus, the receiver must know what symbols are transmitted from the device. To address this issue, the receivers use the preambles that are fixed and known to each other for the DPE. However, preambles are diverse in different protocols. To meet this condition, the platform must attempt to decode each packet using all kinds of protocols one by one, which is a time-consuming and cumbersome procedure.

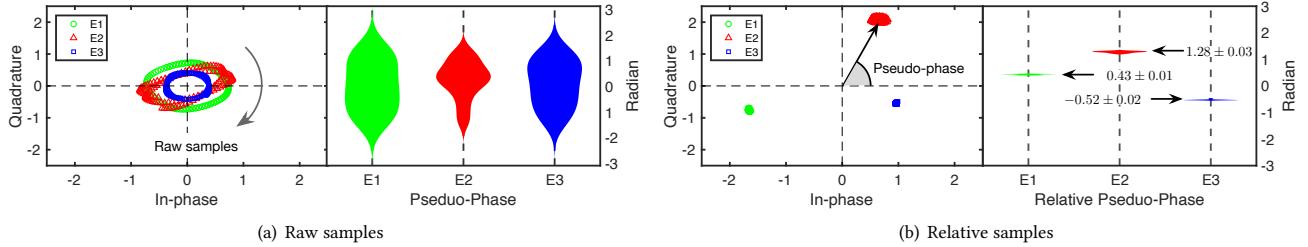


Figure 8: An RF Signal Transmitted from a Static LoRa Device. The signal is captured by three array elements (i.e., E1 ~ E3), each of which acquires 180 samples. (a) On the left, the raw samples are shown in the constellation. These samples rotate as a function of time because the LoRa protocol adopts CSS as its shift-keying scheme, resulting in three loops. On the right, we show the distribution of the samples' pseudo-phases (i.e., the angles of samples). (b) Similarly, we show the relative samples in the constellations and their distribution in pseudo-phase on the left and right respectively. Compared with raw samples, the relative samples collapse into three extremely small clusters in the constellation and remain consistent in the distribution of pseudo-phase.

- **Modulation dependence:** Some modulation techniques, such as phase-shift keying or chirp spread spectrum, represent data by changing the phase of the signal, which leads to extra uncontrollable variables in the pseudo-phase. For example, the LoRa protocol adopts CSS as the shift-keying scheme, which increases the frequency or equivalent phase linearly. Consequently, the pseudo-phase may span over the entire domain and result in an uncertain AoA.

- **Channel dependence:** We assume that CFO and CPO do not exist in Eqn. 10. The CFO not only exists in iArk but also behaves worse than the existing IoT systems because our frontend is fixed to listen at 915 MHz, whereas IoT devices may choose a sub-channel that deviates from the center frequency to reduce signal collisions. Moreover, the RFID Gen 2 protocol requires RFID systems to hop randomly among the 52 sub-channels every 500 ms [1]. Thus, the practical pseudo-phase $\tilde{\theta} = 2\pi\Delta f \cdot t + \theta + \Phi - \Delta\phi$, is derived from Eqn. 8. Thus, the pseudo-phase increases as a function of time, that is, $\tilde{\theta} \sim \Delta f \cdot t$.

- **Device dependence:** True phase θ is hidden inside the pseudo-phase $\tilde{\theta}$ (Eqn. 10). The true phase is the only phase variable related to distance and the key for the AoA estimation. If the pseudo-phase is directly used, then the estimated AoA will contain a large error. To extract θ from $\tilde{\theta}$, transitional solutions usually require a pre-process step similar to that in previous work [53] for estimating Φ in the calibration process. However, this approach is unscalable. We consider a large-sized warehouse where thousands of IoT devices are deployed; thus, measuring Φ for each device is a nearly impossible mission.

The four factors above are protocol-specific. Any ambiguity in one of them may lead to false phase estimation. Thus, past localization solutions must be specifically bound to a device type with only one protocol. To better understand this issue, we show a LoRa signal captured by three array elements, in Fig. 8(a). Ideally, these samples should distribute within a small cluster because the device remains static and the corresponding true phase and pseudo-phase remain constant (see Eqn. 10). In fact, the samples rotate as a function of time, resulting in three loops, due to the presence of CFO and CSS based shift-keying. This physical-layer constraint severely weakens the generalization of past solutions.

5.3 Protocol-free Estimation Algorithm

To address the issue, we develop a novel protocol-free phase estimation algorithm that can address all the dependencies mentioned above [18, 21, 26]. Our intuition is that the main and side channels are driven by the same oscillation such that their signals are well-synchronized and contain identical symbols, the CFO, the CPO, and hardware diversity. Thus, by comparing their signals, we can remove all negative effects. Let $Y_0[t]$ and $Y_{i,j}[t]$ be the received symbol from the side channel and the array element $A_{i,j}$ respectively. $i, j = 0, 1, 2, \dots, 63$. We define the relative sample (i.e., $Z_{i,j}[t]$) at time t as follows:

$$Z_{i,j}[t] = \frac{Y_{i,j}[t]}{Y_0[t]} = \frac{a_{i,j}X[t]e^{j(2\pi\Delta f t + \theta_{i,j} + \phi - \Delta\phi)}}{a_0X[t]e^{j(2\pi\Delta f t + \theta_0 + \phi - \Delta\phi)}} = \left(\frac{a_{i,j}}{a_0}\right)e^{j(\theta_{i,j} - \theta_0)} \quad (11)$$

where $\theta_{i,j}$ and θ_0 are the true phases of the RF signals acquired by the array element $A_{i,j}$ and the side channel, respectively. $Y_k[t]$ and $Y_0[t]$ are defined in Eqn. 8. Correspondingly, the *relative pseudo-phase* of the RF signal received at $A_{i,j}$, which is denoted by $\hat{\theta}_{i,j}$, is computed as follows:

$$\hat{\theta}_{i,j} = \angle Z_{i,j}[t] = \theta_{i,j} - \theta_0 \quad (12)$$

Compared with the traditional pseudo-phase (i.e., $\tilde{\theta}$ in Eqn. 10), the relative pseudo-phase holds the true phase but provides the following key points:

- **Preamble-free:** The transmitted symbol (i.e., $X[t]$) is canceled, which implies that the algorithm does not need to know what symbols are transmitted from the device. The algorithm can estimate the phase using any received symbols more than preambles.
- **Modulation-free:** Similarly, the means of encoding the data on the baseband and modulating onto the carrier becomes irrelevant to the estimation results because $X[t]$ is canceled.
- **Channel-free:** Δf and $\Delta\phi$ are canceled. The platform does not need to know at which frequency the RF source operates and how the data is up- or down-converted. The relative pseudo-phase is totally free of the out-of-sync frequency or phase.

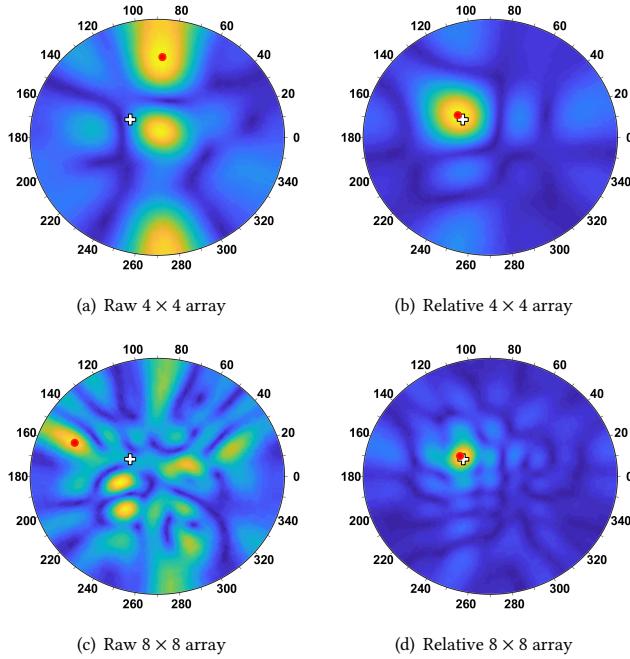


Figure 9: Comparisons of SS Generated with DPE and PPE. The symbol of white cross (+) denotes the ground truth, while the symbol of red circle (●) denotes the peak of the spectrum. The spectrums are computed by using raw pseudo-phase and relative pseudo-phase, which are estimated by DPE and PPE respectively. (a) and (b) are generated by a 4×4 antenna array; (c) and (d) are generated by an 8×8 antenna array.

- **Device-free:** The initial phase shift from the device (i.e., Φ) is canceled. Thus, transmitter calibration (i.e., on IoT devices) is not required.

In summary, the algorithm spontaneously eliminates the heterogeneity of protocols regardless of which type of the preamble is used, which bits are received, and which channel the transmitters operate at. Thus, the algorithm is applicable to any kind of protocol, that is, protocol-free. Importantly, the algorithm can scale up to any number of IoT devices because pre-calibration is unnecessary. To visually understand the protocol-free feature, we show the relative samples of the same LoRa signal in Fig. 8(b) for comparison. As opposed to raw samples, a total of 180 relative samples collected by each element collapse into an extremely small cluster with a stable and the equal angle (i.e., pseudo-phase) instead of a loop. This condition demonstrates that the relative pseudo-phase maintains stability over all the samples without negative affects.

5.4 Put Things Together

The discussion so far focuses on the protocol-free phase estimation (PPE) by virtual of the side channel. Fig. 10 sketches the entire workflow. When gathering 180×64 sample from the hardware (i.e., each of the 64 elements collects 180 samples.), the middleware repeats the following steps.

- **Step 1:** The middleware decides the presence of a packet by comparing the average amplitude of the samples from the side-channel with a user-defined threshold.

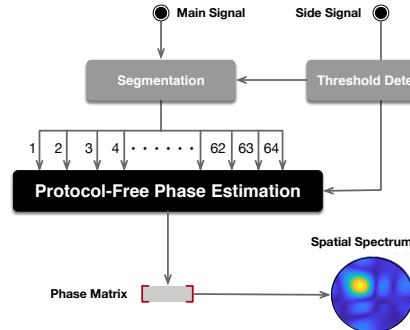


Figure 10: Middleware Workflow

- **Step 2:** If the average value is higher than the threshold, then the main and side signals are divided into 64 segments, each of which contains 180 samples; otherwise, the algorithm skips the following steps and goes back to Step 1.

- **Step 3:** The average relative pseudo-phase across the 180 samples is computed using Eqn. 12 for each array element.

- **Step 4:** The direct AoA of the RF through the spatial spectrum is computed by substituting the relative pseudo-phase into Eqn. 5.

We compare the effectiveness of DPE and PPE in Fig. 9. In the figure, the spatial spectrums are generated with the raw and relative pseudo-phase, estimated by DPE and PPE respectively, as a function of the array size. The two spatial spectrums above are generated by a 4×4 antenna array, while the two below are generated by an 8×8 antenna array. When using the raw pseudo-phase, the direct AoA errors are $(58.7^\circ, 37.2^\circ)$ and $(9.3^\circ, 40.2^\circ)$ for the 4×4 and 8×8 arrays. In contrast, the AoA errors are reduced to $(2.7^\circ, 2.2^\circ)$ and $(1.7^\circ, 0.8^\circ)$ when using the relative pseudo-phase. These results fully demonstrate that the PPE can truly improve the accuracy of AoA estimation significantly.

Discussion. Finally, two points are worth-noting:

- The five IoT protocol has independent preambles. The platform uses the five preamble templates to correlate the incoming signal in turn for the packet detection. If one correlation results exceed a threshold, it confirms the presence of an IoT packet. This is also a very common way in wireless communication to determine the arrival of a packet. The threshold is a user-specific parameter.

- The shortest packet is transmitted from the RFID tags and has an interval of $3300 \mu\text{s}$ among all the IoT protocols, which is $1.5 \times$ longer than the time consumed on the scheduling cycle (i.e., $2112 \mu\text{s}$). Thus, our algorithm can ensure that the device is located at least once during transmission of any kind of packet.

6 LEARNWARE: A FRAMEWORK FOR DEEP TRACKING

This section demonstrates an AI-boosted deep-tracking framework.

6.1 Convolutional Multi-view Stereo

A high-precision AoA can be estimated through the SS across 64 array elements. However, a single AoA result is inadequate to compute the position of an IoT device. In practice, we must obtain at least two AoA results from two different antenna arrays. This process is similar to the binocular stereo (or the multi-view stereo) in

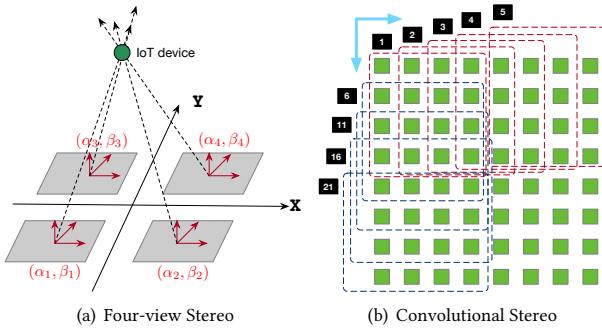


Figure 11: Multi-view Stereo. (a) the antenna array is divided into four 4×4 subarrays; (b) the array is divided into 25 overlapping subarrays.

computer vision, where two (or multiple) cameras displaced horizontally from one another are used to obtain two differing views on a scene. Triangulation is the underlying mechanism behind the multi-view stereo, that is, the transmitter is located at the intersection of two (or multiple) lines along the AoAs. Similarly, we can logically divide the entire antenna array into several sub arrays, each of which can calculate an AoA of the IoT device. The device is finally located at the intersection of the computed AoAs. Fig. 11(a) shows an example wherein the entire array is logically divided into four 4×4 subarrays.

When the incoming signals are phase-synchronized with one another due to multipath, the system perceives distinct incoming signals as one superimposed signal, which results in distortion in the SS and false peaks in $P(\alpha, \beta)$ [44, 52]. To address this issue, we use the *convolutional multi-view stereo* to handle the distortion caused by the multipath. The convolutional multi-view stereo averages the results of multiple overlapping sub arrays. Fig. 11(b) shows a toy example where a 4×4 window is moved from right to left and from top to bottom with one step. We compute the direction by using the 16 antenna elements in each window. As a result, we can obtain 25 AoAs, each of which is assigned a weight. The position of the device is determined by the weighted average. All weights are learned using a neural network, which is described subsequently.

6.2 Deep Tracking Networks

AoA-based localization is widely used in radar and acoustics. However, realizing this task indoors is challenging due to the presence of strong multipath RF propagations. We notice that the final received signal is a weighted superimposition of the signals traveling from multiple paths. The AoA correlation should achieve a peak in the line-of-sight direction. In contrast, the essence of a neural network is to decompose the input signal (e.g. an image) through the approach of weighted averaging where each neuron is assigned a weight. Inspired by this insight, we would like to use the deep neural network to identify the LOS propagation from the final combined signal. Fig. 12 illustrates the structure of the framework. Our framework contains two neural networks, namely, *AoA neural network* (ANN) and *Triangulation neural network* (TNN).

ANN: We leverage the recent success of convolutional neural networks (CNNs), which have demonstrated a major development

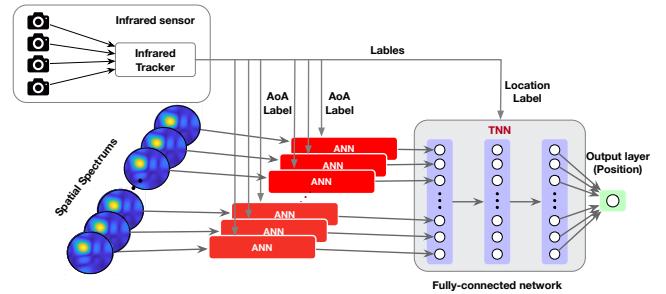


Figure 12: Deep Neural Networks. The overall model can be divided into two main neural networks: AoA neural network (ANN) and triangulation neural network (TNN). ANN is a typical CNN-based deep residual network, whereas TNN is a fully connected network with three layers. The top graph shows the generation of the true AoAs and positions of the transmitter by using infrared motion sensors. They are used to train the two networks.

in object tracking for images and videos [40]. The value of neurons in a CNN can be computed as the convolution of a weighted kernel with the neurons in the previous layer. Our intuition here is that the linear superposition of multipath propagations is extremely similar to the weighted overlay of neurons², whereas the weight kernel is desired to depict such combinations in the SS. The CNN takes the SS as input and outputs the AoA. A few standard CNN designs are widely used across tasks. We choose to use the 18-layer ResNet [20], which uses residual connections across different layers. The final layer is fully connected for regression. The ANN is a general neural network used to identify the AoA from an SS regardless of the spectrum generated by using a logical or a physical antenna array. In iArk, we have 25 logical subarrays. Thus, 25 ANNs are constructed and trained separately. In the example shown in Fig. 9, the AoA errors are further decreased by ANN to $(2.05^\circ, 0.8^\circ)$ and $(1.3^\circ, 0.4^\circ)$ when using the 4×4 and 8×8 antenna arrays respectively. The ANN provides over 50% extra improvements compared with the naive peak based approach.

TNN: We construct another fully-connected neural network including three layers to perform the triangulation. The TNN takes the 25 AoAs estimated by the previous ANNs as the input. Essentially, the network determines the weighted averages of the input AoA values, whereas the weights are automatically learned by training. Two neural networks are not combined into one because a single neural network may cause learning to fall into signature-based recognition. Consequently, the position cannot be correctly predicted if the device is located at a new or distant position that is not sampled in the training phase. Instead, we build the connection between the RF signal and the AoA rather than the position in the ANN. We only need a relatively small number of training data around the antenna array; however, the proposed framework can predict the distant transmitter. We aim to let the network learn how to identify the LOS AoA in the presence of severe multipath effects through the ANN.

²The in-phase and quadrature components of the complex signal can be viewed as the values from two color channels.

Training data. We need to collect two kinds of labels (i.e., AoA and position) for the ANN and the TNN respectively. Thus, we gather the true positions of transmitters by using OptiTrack [10]. The OptiTrack system can track the center of any object with an accuracy of $20\mu\text{m}$ using infrared (IR) cameras. We attach an infrared marker on the antenna of each target device and measure its location via OptiTrack 120 times every second. The AoA relative to the subarray can be computed by the collected position.

7 IMPLEMENTATION

After a two-year effort, we successfully built an entire prototype of iArk with the hardware, middleware, and learnware.

Hardware Settings: Fig. 2 shows the RF frontend. We use the microstrip technique to fabricate the antenna array on a printed circuit board (PCB) for avoiding signal attenuation on the board. The PCB is composed of the substrate of RT/duroid 5880. Each array element has 0.5 dBi gain, 0.5 dB flatness, and 52° beamwidth. The model of high-speed RF switches is BGS18GA14 [2] from Infineon Technologies. We use a USRP 2950 software defined radio (SDR) from NI [6] to build the frontend. The SDR contains two stand-alone I/O interfaces, which are used to build the main and side channels. The RF frontend has a bandwidth of 140 MHz, which covers the entire UHF of $860 \sim 960$ MHz. We also develop a debugging tool to facilitate hardware testing (Fig. 13(b)). The backend runs at a high-performance PC equipped with an Intel CPU Xeon E5-2620 and an NVIDIA GTX 1080Ti GPU.

Middleware Settings: We test five types of IoT devices, namely, ImpinJ Monza QT4 RFID tags [9], iGi XBee Pro 900 HP (Zigbee) [3], Dragino LoRa Shields (LoRa) [8], DFRobot SIMC7000C shields (NB-IoT) [4] and PyCom Sipy chips (Sigfox) [5]. They are shown in Fig. 2. We use a Zigbee device for the following evaluation by default unless otherwise noted. In addition, an ImpinJ 420 reader is utilized to activate RFID tags.

Learnware Settings: As mentioned, we use the OptiTrack system [10] to collect the ground truth. We collect an entirely diverse dataset engaged in the five types of IoT devices. The dataset contains 270,938 samples, of which 84,280 samples are used to train the two networks, and the remaining are utilized for testing. We release iArk’s code and dataset in [11]. The samples are nearly uniformly distributed in the 3D space. The longest duration of continuous testing spans across 48 h. The ANN and the TNN are trained using the ADAM optimizer with a learning rate of 0.01 and an epoch size of 200. Residual connection and batch normalization are adopted in the ANN to benefit the training. The ANN configuration refers to the 18-layer setting shown in Table 1 in [20]. The training is performed in the GPU and takes approximately 1 h, while the testing can be accomplished using the CPU. To save on cost and time, the training workload is advised to be performed in the cloud in practice.

8 EVALUATION

We empirically evaluate the performance of the iArk prototype by conducting experiments in our office of $10 \times 20 \text{ m}^2$, which currently accommodates 20 people. Fig. 13(a) shows the scenario, which is full of different types of indoor reflectors including tables, chairs, computers, and metal obstacles.



Figure 13: Experimental Setup. (a) Evaluation scenario and OptiTrack system and (b) Screenshot of our debug tool.

8.1 Accuracy of AoA Estimation

First, we evaluate the capability of iArk to estimate AoA by using the antenna array. The transmitting device is placed 8 m away from the antenna array. We place the device at different angles and estimate the AoA along the direct path. The ground truth is calculated on the basis of the physical locations of the device and the antenna array.

■ **Angular Accuracy.** Fig. 14(a) plots the CDFs of the AoA estimation errors as determined by the peak of the SS and the ANN. We observe the following findings. (a) If the SS is used, then the median errors of the azimuthal (i.e., α) and elevation (i.e., β) angles are 1.87° and 0.77° , and the 90 percentile errors are 5.1° and 1.87° , respectively. (b) If the ANN is used, then the median errors are 1.5° and 0.46° , and the 90 percentile errors are 3.57° and 1.13° in the two angles, respectively. We see 20% and 40% improvements in the estimation. In either case, the azimuthal angle is better than the elevation angle. This phenomenon may be attributed to the smaller searching space of the elevation angle (i.e., $0^\circ \sim 90^\circ$) than that of the azimuthal angle (i.e., $0^\circ \sim 360^\circ$).

■ **Impact of Array Size.** We examine the impact of array size on accuracy. We select the signals from the adjacent 2×2 , 4×4 , and 8×8 antenna elements to estimate the AoA. Fig. 14(b) plots the errors of AoA estimation for the three cases. When the SS is used directly, the median errors of the azimuthal angle are 8.5° , 2.6° and 1.8° ; those of the elevation angle are 5.5° , 2.3° and 0.7° . By contrast, when ANN is used, the median errors of the azimuthal angle are 22.1° , 1.5° and 1.4° ; those of the elevation angle are 2.8° , 0.5° and 0.4° . Accuracy is evidently improved when additional elements are used because the LOS paths are consistent with the ground truth in different positions, but the NLOS paths are different. This condition is equivalent to increasing the spatial diversity. Only 6% improvement is obtained by the 8×8 array compared with the 4×4 array, but the manufacturing cost is nearly quadruple. A trade-off between accuracy and cost should be made with respect to the practical demand. Here, we choose to use logical 4×4 arrays as described above.

■ **Comparison with State of the Art.** We compare the results of iArk and the recent work (i.e., SWAN) in Fig. 14(c). SWAN extends the commercial Wi-Fi system to an antenna array through RF switches. Given that SWAN works for 2D localization, the figure only plots the comparison of the azimuthal results.³ The average error of SWAN is 3.98° with a standard deviation (SD) of 2.45° , whereas the error of iArk is $1.68 \pm 0.28^\circ$. iArk performs significantly

³Note that the results of SWAN shown in Fig. 14(c) are provided by Y. Xie [51].

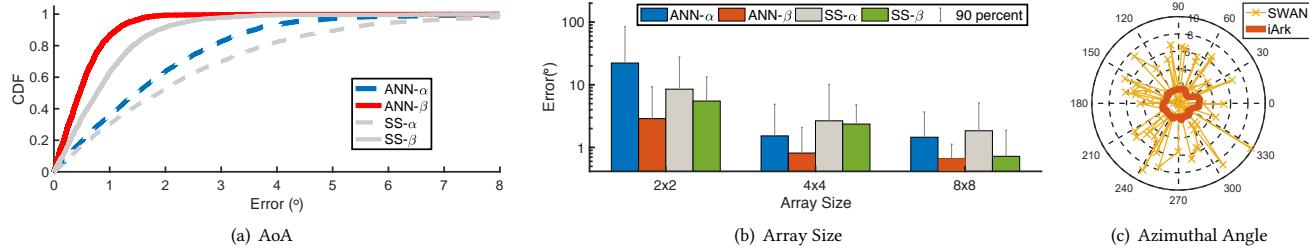


Figure 14: Accuracy in AoA estimation. (a) AoA errors by using ANN and the peak of SS; (b) AoA errors as a function of the array size; (c) comparison to SWAN.

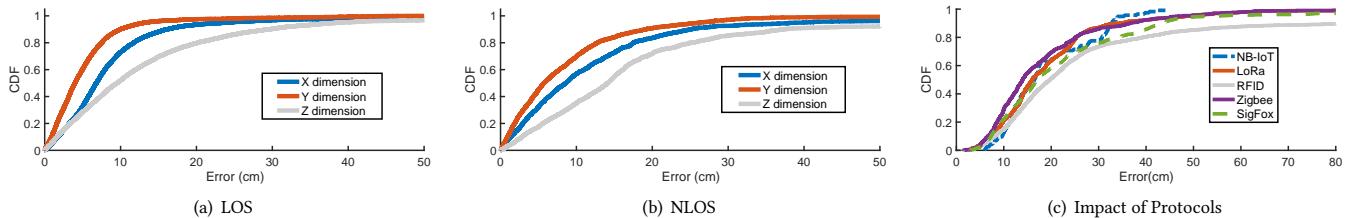


Figure 15: Accuracy in AoA Localization. Localization errors in the (a) LOS and (b) NLOS settings; (c) localization errors regarding protocols.

better than SWAN in all directions. In particular, iArk is uniformly accurate across all the test directions and has a maximum error of 2.51° . It can provide a 360° field of view at azimuth with high accuracy. The SWAN also adopts a switched antenna array, which is manipulated by an Arduino with a scheduling delay of $29.5\mu s$. By contrast, our system uses the FPGA for scheduling with a delay of $3\mu s$. Thus, iArk can acquire more samples. The superiority of iArk can also be attributed to the precise PPE.

8.2 Accuracy of 3D Tracking

We then focus on the developed 3D tracking primitive and evaluate its accuracy across three dimensions. The OptiTrack systems are set to track the target accurately only when it moves within an $8 \times 8 \text{ m}^2$ area in the room. Given that OptiTrack provides the ground truth in our experiment, we move the target in an area that is 1 m away from the antenna array. Thus, the minimum separation between the antenna array and the device in this experiment is 1 m and the maximum separation is approximately 9 m. The influence of distance is discussed later.

Table 2: Comparison with State of the Art

Systems	Model	Target	Location (3D)		AoA	
			50^{th}	90^{th}	50^{th}	90^{th}
WiTrack [13]	ToF	Passive	22 cm	62.9 cm	11.5°	37.9°
ArrayTrack [52]	AoA	Wi-Fi	23 cm	80 cm	-	-
LTEye [26]	SAR	LTE	61 cm	104 cm	6.9°	12.2°
SWAN [51]	AoA	Wi-Fi	45 cm	65 cm	2.6°	5.7°
mD-Track [50]	Hybrid	Wi-Fi	36 cm	71 cm	3.3°	12°
xArray[7]	Beamform	RFID	-	150 cm	-	-
RFind [33]	UWB	RFID	1.9 cm	4.9 cm	-	-
TurboTrack [30]	UWB	RFID	0.5 cm	1.1 cm	-	-
iArk	AoA+DL	IoT	16.2 cm	35 cm	1.5°	3.5°

¹ The above results are directly from the reported papers respectively except RFind and iArk due to the lack of essential hardwares.

■ **Dimensional Accuracy.** Fig. 15 plots the CDFs of the location error along the X, Y and Z coordinates in the two LOS and NLOS settings. For NLOS, we block the direct paths using a wooden obstacle. iArk’s median location errors for the LOS experiments are 6.8, 3.9, and 9.5 cm along the X, Y, and Z dimensions, respectively. By contrast, the median location errors in the NLOS experiments are 8.7, 5.8, and 14.2 cm along the three dimensions. As expected, the location accuracy in LOS is higher than when the device is behind an obstacle due to the additional attenuation, which reduces the SNR. However, the median error in both settings is small.

■ **Impact of Protocol.** We then evaluate the impact of the protocol on localization accuracy. Fig. 15(c) plots the location error in 3D. The median errors of NB-IoT, LoRa, RFID, Zigbee and Sigfox are 16.4, 16.0, 19.7, 13.9 and 16.9 cm, respectively. The difference in localization error across protocols is less than 6 cm, which demonstrates that the performance of tracking algorithms is completely protocol-free. In particular, the RFID performs relatively poorly because the tags are battery-free, and their signals are weak.

■ **Comparison with State of the Art.** We compare iArk with several state-of-the-art indoor localization systems, listed in Table 2. The table shows that iArk can achieve the best performance in terms of accuracy in localization and AoA estimation for commercial devices. All past systems only support a single type of device (i.e., Wi-Fi receivers or RFID tags), whereas our system can serve all types of IoT devices concurrently. The superiority of our system can be attributed to two main reasons. First, stable RF phase measurements are crucial in the upper-layer algorithm. Our PPE powered by the dual-channel hardware design effectively removes all negatives that may affect the phase. It can provide additional uncontaminated and reliable measured data compared with the previous work. Second, most past systems still use the geometric model, and thus fall short of real-life indoor conditions, especially in the presence of strong

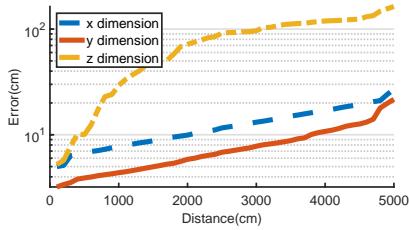


Figure 16: Accuracy vs. Distance

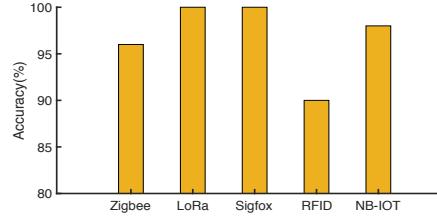


Figure 17: Packet Detection

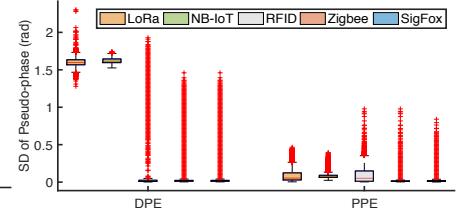


Figure 18: Phase Estimation

multipath propagations. We introduce the deep learning technique to identify the LOS path, which demonstrates its effectiveness.

8.3 Accuracy vs. Distance

We are interested in evaluating the accuracy of iArk as the device moves farther away from the device. We repeat the above-mentioned experiments as a function of distance. As mentioned, OptiTrack requires the device to move within a certain area that is in the LOS of IR cameras. Thus, we move the device and the OptiTrack systems away from the antenna array. The positions with distances to the array greater than 25 m are tested at the adjacent room. Fig. 16 illustrates the localization error of iArk as a function of its distance to the antenna array. The median errors of the estimation for the X, Y, and Z dimensions are shown.

X/Y Dimension. The accuracy along the X-axis (and Y-axis) changes from 5.0 cm (and 3.2 cm) to 27.1 cm (and 21.6 cm) for distances of 1 m to 50 m away from the device. The distance does not affect the accuracy along X and Y coordinates significantly. The decrease is caused by the range-dependent signal attenuation.

Z Dimension. By contrast, the error along the Z dimension is not as good as the other two dimensions. The error increases from 5.2 cm to 163.4 cm. This is because the Z dimension indicates the depth – the distance between the antenna array and the device (Fig. 3). Our 25 logical antenna arrays are deployed on the same side. As a result, the intersection of lines along the determined AoAs is a sharp and long zone along the Z dimension. We argue that this phenomenon is similar to that of the GPS wherein all satellites fly above the Earth's surface (i.e., one-sided). Consequently, the longitude and latitude results of the GPS are significantly better than the elevation result. To reduce the impact of depth, the best deployment approach is to deploy the antenna array at the ceiling. Note that we only present the results of up to 50 m distance, which works for most of the large-sized warehouses. In our future work, we will continue to extend iArk to larger spaces.

8.4 Accuracy vs. Protocol

A key feature of iArk is its capability to remove protocol diversity. In this study, we examine the performance of packet detection and phase estimation with respect to different wireless protocols. We place five types of devices in fixed positions and run iArk. The average packet detection accuracy among five protocols is above 90% as shown in Fig. 17. Such accuracy is quite enough for most applications. Next, we check the effectiveness of PPE compared with the DPE. Ideally, the estimated pseudo-phase should remain stationary because the device locations do not change. Fig. 18 presents the

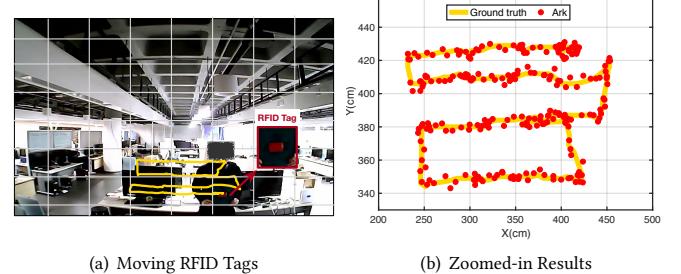


Figure 19: iArk in Real-World Applications. (a) shows the scenario where a person moves an RFID tag attached on a black board and the yellow line indicates the trajectory of the tag. (b) shows the comparison of ground truth and the tracking results.

SD of the pseudo-phase over the 180 samples estimated using the two approaches. From the figure, the SD of PPE follows within 0.15 radians, whereas those of DPE vary in 1.6 radians. In the 32.8 cm wavelength, the two estimation approaches produce potential ranging errors of $0.15/2\pi * 32.8 = 0.78$ cm and $1.6/2\pi * 32.8 = 8.35$ cm. The phase stability of PPE is 90.6% higher than that of DPE. This feature is the key to the high-precision AoA estimation and localization in iArk.

8.5 Case Study

Finally, we qualitatively demonstrate iArk in real-world applications. Fig. 19 presents a scenario of iArk for tracking an RFID tag. The left figure shows that the tag is moved freely in front of the antenna array, while the right figure exhibits the tracking results. iArk can achieve the same level of accuracy reported in the quantitative results above.

9 RELATED WORKS

In this section we survey prior work in location estimation and phase calibration, and describe how these systems relate to iArk.

Localization. Many studies have been conducted on indoor localization. They can be grouped into three main categories.

(1) *Trilateration:* In this category, the receiver estimates the distances to at least three anchors, and then locates itself at the intersection of the multiple spheres. Studies in this category usually require a GHz bandwidth (e.g., > 2 GHz [12, 14, 19, 22, 32, 60, 61]), to estimate the time of flight. For example, RFind [33] and Turbo-Track [30] observe that the RFID tags can respond in a wider band in practice than they claim in the Gen 2 protocol. Inspired by this,

these two works emulate hundreds of MHz bandwidth on tags to locate tags via UWB. Unfortunately, the majority of IoT devices still work at a narrow band of less than 4 MHz. Thus, the trilateration or UWB techniques cannot be used as a general approach for localizing IoT devices in practice.

(2) *Signature*: The second category collects the signal signatures in the entire surveillance region in advance and locates a running IoT device at the position where the signal of the device most matches the stored signature [29, 36, 39, 43, 47, 49, 55, 57]. For example, PinIt [49] captures and extracts the multipath profiles as signatures by using an antenna array, with the intuition that nearby RFIDs experience a similar multipath environment. As a general-purpose platform, iArk can also work for these signature based localization solutions. Particularly, iArk can increase the dimension of signature because it can acquire the signals of IoT devices from different angles, and thereby improve the matching accuracy potentially. In our future work, we will study how could iArk benefits the signature based localization.

(3) *Triangulation*: Similar to ours, the final category leverages an antenna array to estimate the AoA of the wireless signal [13, 21, 23, 25, 37, 42, 51, 52]. For example, DOF [21] identifies unknown radios through the hidden and repeating patterns in their signals and furhter locates them through AoAs. A recent work [17] constructed a phased array on commodity Wi-Fi with three antennas for localization. ArrayTrack [52] and SWAN [51] are the most similar systems to ours. However, iArk is superior to the two works in terms of hardware and middleware. First, the array sizes of ArrayTrack and SWAN are limited to 16 and 12 respectively. On the contrary, our antenna array is 4× and 5× larger than them in size. As demonstrated, the array size is a key factor of accuracy. To the best of our knowledge, ours is the first real large-sized antenna array designed for studying the localization of IoT devices in the community. In particular, ArrayTrack uses 16 standalone ADCs to construct the RF frontend, with a cost that is 8× higher than ours. Second, ArrayTrack and SWAN target the localization of Wi-Fi devices without the challenge of protocol diversity, while one of our contributions is in the pervasiveness derived from the cross-protocol design. Phaser [17] constructs an antenna array by using multiple receiving Wi-Fi radios. The most challenging issue that Phaser addresses is how to overcome the phase difference among the receiving multi-radios caused by the unsynchronized and stand-alone oscillators. iArk contains two receiving channels (or radios), both of which share a single oscillator, thus this issue does not exist in our system. As opposed to Phaser, we take advantage of the phase difference between two channels caused by the difference in locations of two antennas (rather than the unsynchronization) to eliminate the protocol diversity. LTEye [26] can monitor the 3-dimensional physical locations of LET transmitters. Our design is similar to this work but differs in three aspects. First, from the view of hardware, iArk is 250× efficient than LTEye in signal acquisition. iArk uses a real antenna array while LTEye adopts a movable antenna. As a result, iArk only takes 2 ms on the signal acquisitions, but LTEye requires 500 ms at least. Second, from the view of middleware, LTEye also utilizes a static antenna as a reference to overcome the CFO. Beyond this, iArk extends the relative signal to eliminate the protocol diversity including dependences of preamble, modulation, channel and devices. We attempt to resolve the newly

emerging localization issues in IoT even if using a similar technique. Third, from the view of learnware, the localization results of iArk are much more accurate, outperforming LTEye by about 4×, which benefits from the well-designed hardware and the deep learning.

■ **Coexistence**. Many previous works [15, 18, 24, 28, 35, 41, 58, 59] study the coexistence and/or mutual communications of multiple diverse wireless technologies (e.g., Zigbee, WiFi, etc) by designing more efficient protocols. The main goal of these work is to improve the utilization of the spectrum. In contrast, iArk focuses on locating diverse IoT devices regardless which protocols they are using. Thus, iArk can work with these previous works.

10 LIMITATIONS & CONCLUSION

Limitations. iArk leaves room for further investigations, as discussed below: (1) Transfer learning. Neural networks assume that the learned models are static over time. However, in a complex indoor building, the environment is always dynamic, due to the unpredictable movements of people and radio interference. Thus, the propagation paths and distributions of phase values at training and test periods may be significantly different. This difference causes the learned models to fail in the test period if using the data collected at the trained period. Prior work in machine learning has demonstrated that information learned at the early layers of a neural network is used to model the input and mostly independent of the output, whereas the later layers are more specialized. This phenomenon inspires the community to transfer the learning to accommodate dynamic [38, 39]. The basic idea is to recollect a few new training data in the new environment to update the last few layers while maintaining the previous layers unchanged [56]. We plan to adopt the transfer learning to deal with the dynamics in our future work. (2) Scalability. Acting as a sniffer, the platform does not need to know which device transmits the signal or whether a signal collision occurs. The platform neither introduces additional traffics or collisions nor requires cooperation from other devices. The platform simply stores all location results over the phase matrices in the database with a timestamp. Thus, the platform can support any number of devices as long as they could successfully communicate with each other. Thus, our design is essentially scalable for any kind of IoT network with a large number of devices.

Conclusion. This work presents iArk, which is a general-purpose platform for tracking IoT devices across protocols. To the best of our knowledge, this study is the first to propose an all-in-one platform that can track all types of IoT devices. The design of iArk introduces three key innovations. First, it presents a practical hardware design of an $8 \times 8 + 1$ antenna array. Second, a middleware is designed to remove device diversity and protocol varieties. Third, a deep tracking framework is developed to improve the tracking accuracy and stability. The platform provides a wide range of exciting opportunities for developing tracking systems.

ACKNOWLEDGMENTS

Zhenlin An and Qiongzheng Lin are the co-primary authors. Lei Yang is the corresponding author. The research is supported by NSFC for Young Scientists of China (NO. 61972331), NSFC General Program (NO. 61902331), NSFC Key Program (NO. 61932017) and UGC/ECS (NO. 25222917). We thank all the anonymous reviewers and the shepherd, Dr. Eric Rozner, for their valuable comments and helpful suggestions.

REFERENCES

- [1] 2014. EPCglobal Gen2 Specification. www.gs1.org/epcglobal.
- [2] 2016. RF Switch. <https://www.infineon.com/cms/cn/product/rf-wireless-control/rf-switches-spst-dpxt/bgs15ga14/>.
- [3] 2016. Zigbee node. <https://www.digi.com/products/embedded-systems/rf-modules/sub-1-ghz-modules/xbee-pro-900hp>.
- [4] 2017. NB-IOT. https://wiki.dfrobot.com/SIM7000_Arduino_NB-IoT_LTE_GPRS_Expansion_Shield_SKU_DFR0505_DFR0572.
- [5] 2018. Sigfox. <https://pycom.io/product/sipy/>.
- [6] 2018. USRP 2950. <http://www.ni.com/en-us/support/model.usrp-2950.html>.
- [7] 2019. Impinj xarray. <https://www.impinj.com/platform/connectivity/xarray/>.
- [8] 2019. LoRa. <https://www.dragino.com/products/module/item/102-lora-shield.html>.
- [9] 2019. RFID Tag. <https://support.impinj.com/hc/en-us/articles/202756908-Monza-4-RFID-Tag-Chip-Datasheet>.
- [10] 2020. OptiTrack. <https://optitrack.com/>.
- [11] 2020. Source Code. <https://github.com/Anplus/iArk>.
- [12] Fadel Adib, Zachary Kabelac, and Dina Katabi. 2015. Multi-Person Motion Tracking via RF Body Reflections. In *Proc. of USENIX NSDI*.
- [13] Fadel Adib, Zachary Kabelac, Dina Katabi, and Rob Miller. 2014. WiTrack: motion tracking via radio reflections off the body. In *Proc. of USENIX NSDI*.
- [14] Fadel Adib, Zach Kabelac, Dina Katabi, and Robert C Miller. 2013. 3D tracking via body radio reflections. In *Proc. of USENIX NSDI*, Vol. 14.
- [15] Kameswari Chebrolu and Ashutosh Dhekne. 2009. Esense: Communication through energy sensing. In *Proc. of ACM MobiCom*.
- [16] Yuhang Gao, Jianwei Niu, Ruogu Zhou, and Guoliang Xing. 2013. ZiFind: Exploiting cross-technology interference signatures for energy-efficient indoor localization. In *Proc. of IEEE INFOCOM*.
- [17] Jon Gjengset, Jie Xiong, Graeme McPhillips, and Kyle Jamieson. 2014. Phaser: Enabling phased array signal processing on commodity WiFi access points. In *Proc. of ACM MobiCom*. 153–164.
- [18] Shyamnath Gollakota, Fadel Adib, Dina Katabi, and Srinivasan Seshan. 2011. Clearing the RF smog: making 802.11 n robust to cross-technology interference. In *Proc. of ACM SIGCOMM*. 170–181.
- [19] Azril Haniz, Gia Khanh Tran, Kentaro Saito, Kei Sakaguchi, Jun-ichi Takada, Daisuke Hayashi, Toshihiro Yamaguchi, and Shintaro Arata. 2017. A novel phase-difference fingerprinting technique for localization of unknown emitters. *IEEE Transactions on Vehicular Technology* 66, 9 (2017), 8445–8457.
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. of the IEEE CVPR*. 770–778.
- [21] Steven Siying Hong and Sachin Rajsekhar Katti. 2011. DOF: a local wireless information plane. In *Proc. of ACM SIGCOMM*. 230–241.
- [22] Xiaoman Hui and Edwin C Kan. 2019. Radio ranging with ultrahigh resolution using a harmonic radio-frequency identification system. *Nature Electronics* 2, 3 (2019), 125.
- [23] Kiran Raj Joshi, Steven Siying Hong, and Sachin Katti. 2013. PinPoint: Localizing Interfering Radios. In *Proc. of USENIX NSDI*. 241–253.
- [24] Song Min Kim and Tian He. 2015. Freebee: Cross-technology communication via free side-channel. In *Proc. of ACM MobiCom*.
- [25] Swaran Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. 2014. Accurate indoor localization with zero start-up cost. In *Proc. of ACM MOBICOM*.
- [26] Swaran Kumar, Ezzeldin Hamed, Dina Katabi, and Li Erran Li. 2014. LTE radio analytics made easy and accessible. *ACM SIGCOMM Computer Communication Review* 44, 4 (2014), 211–222.
- [27] Ka-Ho Lam, Chi-Chung Cheung, and Wah-Ching Lee. 2017. LoRa-based localization systems for noisy outdoor environment. In *Proc. of IEEE WiMob*. IEEE, 278–284.
- [28] Zhijun Li and Tian He. 2017. WEbee: Physical-Layer Cross-Technology Communication via Emulation. In *Proc. of ACM MobiCom*.
- [29] Hongbo Liu, Yu Gan, Jie Yang, Simon Sidhom, Yan Wang, Yingying Chen, and Fan Ye. 2012. Push the limit of WiFi based localization for smartphones. In *Proc. of ACM MobiCom*. 305–316.
- [30] Zhihong Luo, Qiping Zhang, Yunfei Ma, Manish Singh, and Fadel Adib. 2019. 3D backscatter localization for fine-grained robotics. In *Proc. of USENIX NSDI*. 765–782.
- [31] Yunfei Ma, Xiaonan Hui, and Edwin C Kan. 2016. 3D real-time indoor localization via broadband nonlinear backscatter in passive devices with centimeter precision. In *Proc. of ACM MobiCom*.
- [32] Yunfei Ma and Edwin Chihchuan Kan. 2014. Accurate indoor ranging by broadband harmonic generation in passive NLTL backscatter tags. *IEEE Transactions on Microwave Theory and Techniques* 62, 5 (2014), 1249–1261.
- [33] Yunfei Ma, Nicholas Selby, and Fadel Adib. 2017. Minding the billions: Ultra-wideband localization for deployed rfid tags. In *Proc. of ACM MobiCom*.
- [34] Heinrich Meyr, Marc Moeneclaey, and Stefan Fechtel. 1997. *Digital communication receivers: synchronization, channel estimation, and signal processing*. John Wiley & Sons, Inc.
- [35] Shridhar Mubaraq Mishra, Robert W Brodersen, Stephan ten Brink, and Ravishankar Mahadevappa. 2007. Detect and avoid: an ultra-wideband/WiMAX coexistence mechanism. *IEEE Communications Magazine* 45, 6 (2007), 68–75.
- [36] L.M. Ni, Y. Liu, Y.C. Lau, and A.P. Patil. 2004. LANDMARC: Indoor Location Sensing using Active RFID. *Wireless networks* (2004).
- [37] Dragoș Niculescu and Badri Nath. 2004. VOR base stations for indoor 802.11 positioning. In *Proc. of ACM MobiCom*.
- [38] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.
- [39] Sinno Jialin Pan, Vincent Wencheng Zheng, Qiang Yang, and Derek Hao Hu. 2008. Transfer learning for wifi-based indoor localization. In *Proc. of ACM AAAI workshop*, Vol. 6.
- [40] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, Jongwoo Lim, and Ming-Hsuan Yang. 2016. Hedged deep tracking. In *Proc. of IEEE CVPR*. 4303–4311.
- [41] Hariharan Rahul, Nate Kushman, Dina Katabi, Charles Sodini, and Farinaz Edalat. 2008. Learning to share: narrowband-friendly wideband networks. *ACM SIGCOMM Computer Communication Review* (2008).
- [42] Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. 2013. Avoiding multipath to revive inbuilding WiFi localization. In *Proc. of ACM MobiSys*. 249–262.
- [43] Souvik Sen, Božidar Radunovic, Romit Roy Choudhury, and Tom Minka. 2012. You are facing the Mona Lisa: Spot localization using PHY layer information. In *Proc. of ACM MobiSys*. 183–196.
- [44] Tie-Jun Shan, Mati Wax, and Thomas Kailath. 1985. On spatial smoothing for direction-of-arrival estimation of coherent signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33, 4 (1985), 806–811.
- [45] Clayton Shepard, Hang Yu, Narendra Anand, Erran Li, Thomas Marzetta, Richard Yang, and Lin Zhong. 2012. Argos: Practical many-antenna base stations. In *Proc. of ACM MobiCom*. ACM, 53–64.
- [46] Chong Wang, Hongyi Wu, and Nian-Feng Tzeng. 2007. RFID-based 3-D positioning schemes. In *Proc. of IEEE INFOCOM*.
- [47] He Wang, Souvik Sen, Ahmed Elghohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. 2012. No need to war-drive: Unsupervised indoor localization. In *Proc. of ACM MobiSys*. 197–210.
- [48] Jue Wang, Fadel Adib, Ross Knepper, Dina Katabi, and Daniela Rus. 2013. RF-compass: robot object manipulation using RFIDs. In *Proc. ACM MOBICOM*.
- [49] Jue Wang and Dina Katabi. 2013. Dude, where's my card?: RFID positioning that works with multipath and non-line of sight. In *Proc. of ACM SIGCOMM*.
- [50] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. 2018. mD-Track: Leveraging Multi-Dimensionality in Passive Indoor Wi-Fi Tracking. *arXiv preprint arXiv:1812.03103* (2018).
- [51] Yaxiong Xie, Yanbo Zhang, Jansen Christian Liando, and Mo Li. 2018. SWAN: Stitched Wi-Fi ANtennas. In *Proc. of ACM MobiCom*.
- [52] Jie Xiong and Kyle Jamieson. 2013. ArrayTrack: A Fine-Grained Indoor Location System. In *Proc. of USENIX NSDI*. 71–84.
- [53] Jie Xiong, Karthikeyan Sundaresan, and Kyle Jamieson. 2015. Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization. In *Proc. of ACM MobiCom*. 537–549.
- [54] Lei Yang, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. 2014. Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices. In *Proc. of ACM MobiCom*. 237–248.
- [55] Zheng Yang, Chenshu Wu, and Yunhao Liu. 2012. Locating in fingerprint space: wireless indoor localization with little human intervention. In *Proc. of ACM MobiCom*. 269–280.
- [56] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems*. 3320–3328.
- [57] Moustafa Youssef and Ashok Agrawala. 2005. The Horus WLAN location determination system. In *Proc. of ACM MobiSys*. 205–218.
- [58] Xinyu Zhang and Kang G Shin. 2013. Gap sense: Lightweight coordination of heterogeneous wireless devices. In *Proc. of IEEE INFOCOM*.
- [59] Yifan Zhang and Qun Li. 2013. Howies: A holistic approach to zigbee assisted wifi energy savings in mobile devices. In *Proc. of IEEE INFOCOM*.
- [60] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. 2018. Through-wall human pose estimation using radio signals. In *Proc. of IEEE CVPR*. 7356–7365.
- [61] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. 2018. RF-based 3D skeletons. In *Proc. of ACM SIGCOMM*. 267–281.