

# MAIDE: Augmented Reality (AR)-facilitated Mobile System for Onboarding of Internet of Things (IoT) Devices at Ease

HUANLE ZHANG, University of California, Davis

MOSTAFA UDDIN, Peraton Labs

FANG HAO, Nokia Bell Labs

SARIT MUKHERJEE, Nokia Bell Labs

PRASANT MOHAPATRA, University of California, Davis

Having an efficient onboarding process is a pivotal step to utilize and provision the IoT devices for accessing the network infrastructure. However, the current process to onboard IoT devices is time-consuming and labor-intensive, which makes the process vulnerable to human errors and security risks. In order to have a streamlined onboarding process, we need a mechanism to reliably associate each digital identity with each physical device. We design an onboarding mechanism called MAIDE to fill this technical gap. MAIDE is an Augmented Reality (AR)-facilitated app that systematically selects multiple measurement locations, calculates measurement time for each location, and guides the user through the measurement process. The app also uses an optimized voting-based algorithm to derive the device-to-ID mapping based on measurement data. This method does not require any modification to existing IoT devices or the infrastructure and can be applied to all major wireless protocols such as BLE, and WiFi. Our extensive experiments show that MAIDE achieves high device-to-ID mapping accuracy. For example, to distinguish two devices on a ceiling in a typical enterprise environment, MAIDE achieves ~95% accuracy by measuring 5 seconds of Received Signal Strength (RSS) data for each measurement location when the devices are 4 feet apart.

CCS Concepts: • **Human-centered computing** → **Accessibility systems and tools**; **User interface design**.

Additional Key Words and Phrases: IoT, onboarding, AR, mobile, wireless, RSS

## ACM Reference Format:

Huanle Zhang, Mostafa Uddin, Fang Hao, Sarit Mukherjee, and Prasant Mohapatra. 2021. MAIDE: Augmented Reality (AR)-facilitated Mobile System for Onboarding of Internet of Things (IoT) Devices at Ease. *ACM Trans. Internet Things* 1, 1, Article 1 (January 2021), 21 pages. <https://doi.org/10.1145/3506667>

## 1 INTRODUCTION

The Internet of Things (IoT) has increasingly been deployed in various environments such as homes [8], hospitals [26], industry [33], to help improve convenience of life, energy efficiency, safety and security, and product quality. In order to utilize the IoT devices, one crucial yet somewhat neglected step is to have an efficient *onboarding* process that can initialize and provision the devices for accessing the network infrastructure. In the current practice, the process of onboarding IoT

---

Authors' addresses: Huanle Zhang, [dtczhang@ucdavis.edu](mailto:dtczhang@ucdavis.edu), University of California, Davis, Davis, California; Mostafa Uddin, Peraton Labs, Basking Ridge, New Jersey, [mostafa.uddin@peratonlabs.com](mailto:mostafa.uddin@peratonlabs.com); Fang Hao, Nokia Bell Labs, Holmdel, New Jersey, [fang.hao@nokia-bell-labs.com](mailto:fang.hao@nokia-bell-labs.com); Sarit Mukherjee, Nokia Bell Labs, Holmdel, New Jersey, [sarit.mukherjee@nokia-bell-labs.com](mailto:sarit.mukherjee@nokia-bell-labs.com); Prasant Mohapatra, University of California, Davis, Davis, California, [pmohapatra@ucdavis.edu](mailto:pmohapatra@ucdavis.edu).

---

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](mailto:permissions@acm.org).

© 2021 Association for Computing Machinery.

2577-6207/2021/1-ART1

<https://doi.org/10.1145/3506667>

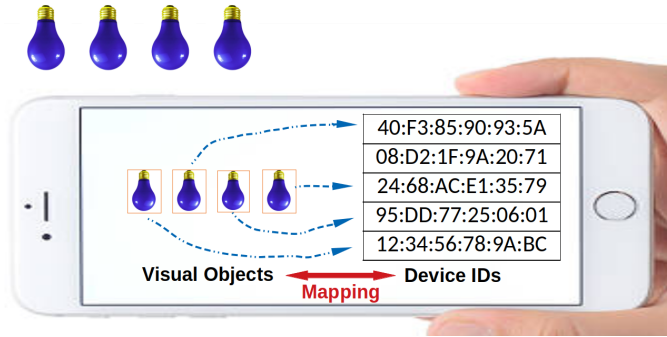


Fig. 1. Onboarding problem of associating visual objects with device IDs.

devices is often time-consuming and labor-intensive, which becomes a barrier to streamlined IoT adoption and deployment [11, 24]. In addition, the complexity of manually deploying a large number of devices makes the system vulnerable to human errors and prone to security risks.

The current manual onboarding process can be illustrated in the following example. Consider a scenario where an enterprise has acquired many smart light bulbs and installed them on ceilings, walls, or floors. These devices are equipped with wireless communication modules such as Bluetooth Low Energy (BLE), WiFi, and Zigbee. In order to allow the system administrators or operators to control these devices, we need to know the device ID (MAC address or physical address) of each light bulb. IoT devices usually broadcast their identity using periodic advertisements (or beacons) that contain human-readable manufacturer names and device types. These attributes can help to separate different types of devices (e.g., light bulbs vs. thermostats), or devices from different manufacturers. However, it is difficult to know (physically) which device has what device ID based on only the advertisement packets. For example, advertisements from a group of light bulbs from the same manufacturer are identical except for the individual MAC addresses (i.e., device ID); and there is no obvious way of mapping a device ID back to its origin device from the received advertisements only. Ideally, we would like to have a simple methodology for such mapping. For example, as Figure 1 depicts, we hold a smartphone camera in front of a group of identical devices and an app running in the phone maps the visual objects (images or icons) to the device IDs received from their advertisements. Since the mapping of visual objects on screens to physical devices is straightforward, the mapping of device IDs to physical devices is achieved with this smartphone system.

To onboard these light bulbs manually, we may either try to find the MAC address on the original package of each device and enter them into the onboarding system one by one, or we can try to onboard each light bulb one at a time by turning it on/off and verifying which device is under control. Such a manual process is extremely tedious and error-prone, and is very inefficient when the number of devices is large. In addition, for devices that do not give visual feedback about its operational status, e.g., sensors that do not show on/off status, it can be difficult to verify their device IDs without testing each of them in an isolated environment free of interference from other devices.

In order to have a streamlined process to onboard IoT devices at large numbers, we need a mechanism to register each device to the infrastructure based on its unique digital identity (i.e., MAC or physical address), and to associate each digital identity with each physical device. Knowing this information enhances safety [11] and usability [7] in interacting with surrounding IoT devices. In this paper, we refer to such a process as *augmented onboarding*. Several practical constraints and

considerations make designing an augmented onboarding system challenging: (1) the onboarding process is suitable for existing IoT devices without any modifications; (2) the onboarding process works for various wireless protocols which means that only information commonly available to wireless protocols are used; (3) no infrastructure and specialized devices are required; and (4) the process is automated as much as possible, and provides unambiguous instructions if requiring human involvement.

We address these system requirement challenges. We design a practical procedure called Mobile Augmented onboarding of IoT Device at Ease (MAIDE), which uses Commercial Off-The-Shelf (COTS) smartphones without additional hardware or infrastructure support, and exploits widely-available RSS profiles of wireless packets (e.g., WiFi and BLE) to associate the visual presence of IoT devices to their digital device IDs. Our main contribution is the design of an Augmented Reality (AR)-facilitated phone app that (1) directs the user to carefully selected locations corresponding to deployed target devices, (2) measures and records RSS profile at the locations, and (3) then uses an optimized matching algorithm to associate each device ID to the target device. Furthermore, we have implemented a smartphone app to run extensive evaluations of the onboarding procedure under various circumstances.

We have explored the possibility of using RSS to onboard devices in our preliminary work, called AIDE [35]. Despite having promising results, AIDE has no systematic design and implementation to find appropriate measurement locations and measurement duration. In comparison, MAIDE is an AR-facilitated system that systematically selects multiple measurement locations relative to the target device locations, calculates the measurement time needed based on the distance from each target device to the phone and the projected accuracy for each measurement, and then gives clear moment-by-moment instructions to the user where to move next and how long to measure. In addition, we have conducted a more comprehensive performance evaluation to investigate the onboarding accuracy under various practical parameters or factors (e.g., measurement time, interference from non-target devices).

We implement a prototype of MAIDE in two smartphone models (Google Pixel 2 and Huawei Mate 20). The overview of MAIDE is described in Section 2. We deploy BLE devices at various sites (e.g., conference rooms, corridors) and arrange them into different topologies (e.g., grid, line). Our extensive experiments show that MAIDE achieves high device-to-ID mapping accuracy. To distinguish two devices on a ceiling in the three different typical enterprise environment settings that we have experimented with, MAIDE achieves 100% accuracy by measuring 6 seconds' RSS data for each measurement location when the devices are 6 feet apart and by measuring 1 second's data for each measurement location when the devices are 10 feet apart.

## 2 SYSTEM DESIGN

We believe that for an onboarding procedure like MAIDE to be practical, it should abide by the following four guidelines: (1) There should be no software or hardware modifications to IoT devices. As billions of IoT devices have already been manufactured, any requirement for modifications to devices makes onboarding system impractical. (2) It should have capability to support mainstream wireless communication standards, such as WiFi and Bluetooth. To be protocol independent, it is desired to rely only on the common information available in all these wireless standards. (3) No infrastructure support and COTS smartphones are used as receivers to develop a scalable approach of inferring the device identities (e.g., MAC address) of the newly deployed IoT devices (i.e., target devices). (4) Human involvement should be minimized. And in cases this cannot be avoided, there should be clear instruction and feedback to reduce the chance of human error.

Figure 2 shows the system diagram of MAIDE, which consists of three stages. Many of the key design decisions for these stages are based on the above design guidelines. (1) The *Filtering* stage.

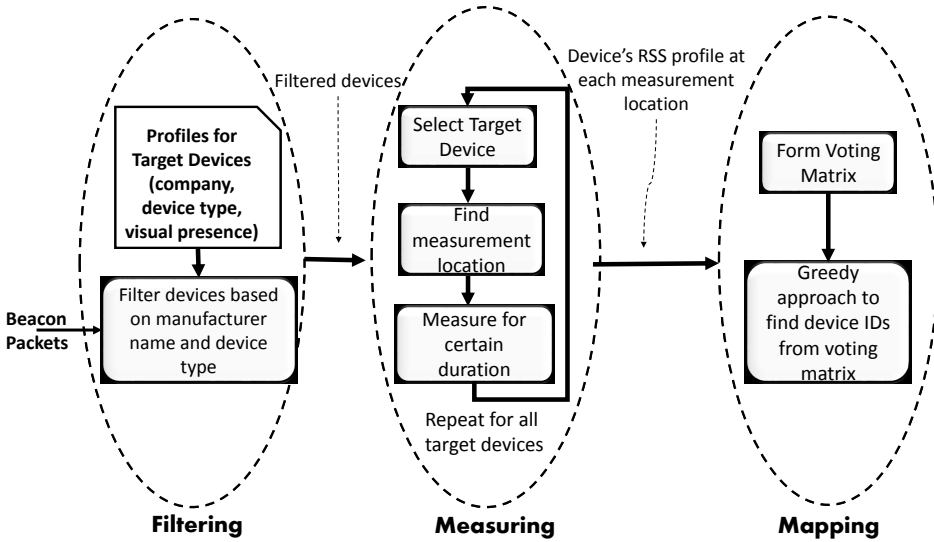


Fig. 2. System diagram of MAIDE. It consists of three main stages: filtering, measuring, and mapping.

We collect broadcast advertisement messages (i.e., beacon packets) from surrounding devices. From these beacon packets, we select the possible set of devices whose identities are potential candidates for the target devices. Thus, we narrow down the search-space for identifying the device ID of the target devices. (2) The *Measurement* stage. We measure the signal features of the received advertisement messages from the selected candidate devices at a unique measurement location for each target device. (3) The *Mapping* stage. We use a novel voting-based algorithm based on the measurement at different locations to infer the identities of the target devices.

**Filtering.** Once IoT devices (i.e., target devices) are deployed and powered on, they start to emit beacon packets. In addition to the target devices, there could be other devices (i.e., non-target devices) in proximity as well. During onboarding procedure, MAIDE receives beacon packets from both the target and non-target devices. Given the number of target devices and their corresponding locations in the physical world, MAIDE needs to infer the identity of the target devices from the beacon packets.

A beacon packet typically contains a *device name* field and a *physical address* field [6]. We observe that the *device name* field may explicitly have the manufacturer name and/or the device type (e.g., light bulb, power socket). Although MAIDE can work for the case where different devices are mixed together, such additional information can narrow down the searching and matching space and hence reduce measurement time and increase accuracy. More specifically, we consider the following three cases: (1) Both manufacturer name and device type of the target devices are unknown. This is the baseline case, in which MAIDE collects beacon packets from both target and non-target devices, and searches through all devices to infer the device IDs of the target devices. (2) If the device type information of the target devices is known in advance, MAIDE can filter out all non-target devices that advertised a different device type in the beacon packet compared to the target device. (3) Similarly, knowing the manufacturer name of the devices helps MAIDE to filter non-target devices that advertised different manufacturer names compared to the target devices.

In our onboarding procedure, all target devices should be of the same type from the same manufacturing company if they are known. In the case of different device types and/or manufacturing



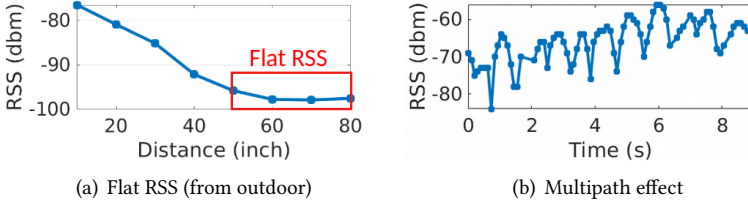


Fig. 3. Signal constraints. (a) Flat RSS beyond some distance; (b) Noisy RSS due to the multipath effect.

companies, MAIDE groups them separately based on the manufacturer name (if known) and/or the device type (if known), and then runs an independent onboarding procedure for each group. For instance, if we need to onboard 4 light bulbs from Samsung and 3 light bulbs from LG, MAIDE conducts two independent onboarding procedures: one for Samsung light bulbs and the other one for LG light bulbs, despite these devices having the same device type. However, in the case that the manufacturer name is unknown for the above light bulbs, then MAIDE uses a single onboarding procedure for all 7 light bulbs.

**Measurement.** In the *measurement* stage, we collect the signal features from the advertisement messages of the selected devices from the *filtering* stage. We conduct one measurement for each target device. We develop a novel scheme to determine the suitable measurement location and the duration of each measurement, where each measurement location is physically mapped to a target device. We elaborate the measurement stage in Section 3.

**Mapping.** The *mapping* stage combines the signal features measured from multiple *measurement locations* to determine the mapping of each device ID for each target device. We develop a voting-based algorithm where each device ID obtains a vote for a measurement location corresponding to a target device. Based on the vote, we map which device ID corresponds to which target device. We elaborate the mapping stage in Section 4.

### 3 MEASUREMENT STEP

There are a number of signal features such as Channel State Information (CSI) [4, 31], Angle of Arrival (AoA) [12], Time of Arrival (ToA) [32] and Received Signal Strength (RSS). Among them, we find RSS more appropriate for the measurement step. For example, CSI and AoA require specialized hardware. Moreover, CSI is only available in WiFi communications. By comparison, COTS smartphones provide RSS information for each received WiFi and BLE packet. It is known that the signal strength from a nearby device tends to be greater than that from a far-away device. Based on that, we can select the device ID that has the strongest RSS at the closest measurement location of a target device. However, such a naive approach fails to work in practice due to signal constraints and physical constraints of the measurement. These two challenges are common for all RSS-based systems.

#### 3.1 Measurement Challenges

Despite the benefit of readily available, RSS-based measurements bring many challenges, including different transmission powers, distances, and the multipath effect.

- (1) Different devices have different transmission powers. Assume that we have two devices of the same type (device ‘A’ and ‘B’) side-by-side, and their transmission powers differ because one has (device ‘A’) higher battery capacity than the other (device ‘B’). An increase in transmission powers increases the RSS value of the received signal. Given the proximity of device ‘A’ and

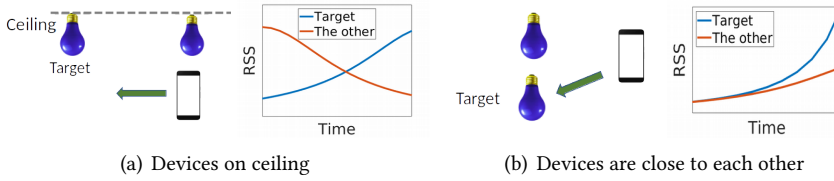


Fig. 4. Physical constraints. (a) Devices are not approachable; (b) Device placement introduces difficulty.

'B', even if we measure RSS of both devices at the location closer to device 'B', we may see a higher RSS value of device 'A' compared to device 'B'. Thus we cannot rely on the absolute RSS value to infer the proximity of devices.

- (2) Beyond a certain distance, changes in RSS are indistinguishable. Figure 3(a) shows a trace (collected outdoor at open-space on the top of Crawford Hill, New Jersey) in which the RSS does not decrease much beyond  $\sim 50$  inches. It means that beyond a certain distance it is challenging to create enough signal strength contrast among multiple devices.
- (3) RSS data in indoor environments are noisy because of the multipath effect. Figure 3(b) shows a trace of RSS when we walk with a receiver directly toward a transmitter located at 80 inches away. Although the RSS increase is the general trend, the data fluctuates significantly. Due to the multipath effect, measuring at a larger distance may show a higher RSS value compared to a shorter distance from the target device. Therefore, without proper techniques to combat the multipath effect, the accuracy of onboarding based on RSS degrades.

In addition to the above signal constraints, the complex layouts of indoor environments also introduce difficulty in achieving precise RSS measurements.

- (1) Devices might not be approachable. For example, if devices are deployed on a ceiling, we cannot get very close to the target devices because of their height. Figure 4(a) shows a case that devices are deployed on the ceiling. In this case, we can move in a side-to-side way, and look at the difference in the trend of their RSS value change.
- (2) Device placement introduces difficulty. Figure 4(b) illustrates one case in which two devices are close to each other. When we move closer to the target device, we can see both RSS values increase. In this case longer measurement time may be needed to differentiate the RSS values. Nevertheless, in the ideal case, we can still distinguish the target device because the RSS increase rate of the target device is greater than the other device.

Before describing the measurement procedure, we first present our measurement technique in mitigating the multipath effect. Afterward, we propose a mathematical model to estimate the suitable measurement location and the duration of each measurement. Finally, by putting them together, we develop the methodology of conducting measurements using augmented reality features.

### 3.2 Mitigating MultiPath Effect

The multipath effect can have both constructive (multipath components are in phase) and destructive (multipath components are out of phase) interference effect on RSS measurements. In such a phenomenon, for the constructive case we see relatively higher RSS values, and relatively lower RSS values for the destructive case. Therefore, we move our phone in a circular way (i.e., local movement) with our hand when we collect RSS data. By doing this, we average RSS (spatially) within a small region, and thus we mitigate the multipath effect in our measurement. Note that

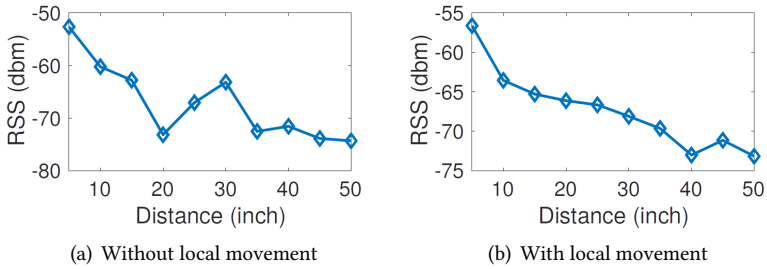


Fig. 5. To mitigate the multipath effect, we move our phone in a circle way. (a) and (b) plots one trace with and without local movement respectively.

the radius of the circular movement has to be at least 2.5 inches, which is half of the wavelength (i.e.,  $\lambda = c/f = 3 \times 10^8 / 2.4 \times 10^9$  meters  $\approx 2.5$  inch). Thus, we can have measurements across full wavelength. To show the effectiveness of our local movement method, we measure RSS at different distances from a transmitter and average RSS data at each location. Figure 5(a) and Figure 5(b) plot the results with and without local movement, respectively. We conduct this experiment at the same locations in order to keep the same environmental factors. Our result clearly shows that our local movement method results in a smoother and more consistent RSS curve over distance in indoor environments. Please note that we do not propose a new RSS-to-distance model. Instead, we propose a measurement method that results in a smoother RSS-to-distance curve. Ideally, for each target device, the RSS value at its target measurement location is higher than the other measurement locations (refer to Section 3.3 for the details of measurement locations). Nonetheless, our mapping algorithm (Section 4) considers all devices at all measurement locations, which is robust against multipath and Non-Line-of-Sight (NLOS).

### 3.3 Measurement Location and Duration Estimation

The basic idea of finding a suitable measurement location for a target device is to create the highest possible signal strength contrast between the beacons received from the target device and its nearest neighbor device. Consider two adjacent target devices (device 1 and device 2). Given the log distance path-loss model [9], the following equation shows the relation between signal strength contrast and the distances at the measurement locations for the target, device 1.

$$RSS_{11} - RSS_{12} = 10 \cdot \alpha \cdot \log(d_{12}/d_{11}) \quad (1)$$

where  $RSS_{11}$  and  $RSS_{12}$  represents the RSS of the beacon packet received from device 1 and device 2 respectively. Similarly,  $d_{11}$  and  $d_{12}$  represents the propagation path distance from the measurement location to each device.  $\alpha$  is the path loss exponent. The value of  $\alpha$  depends on the surrounding environment. In a typical industrial factory floor plan with large rooms, the  $\alpha$  value is between 1.6-3.3[13].

Given the small and constant  $\alpha$  value in a particular environment, the only way to create large contrast between  $RSS_{11}$  and  $RSS_{12}$  is to increase the distance ratio, i.e.,  $d_{12}/d_{11}$ . This means to make the measurement location closer to the target device compared to other nearby devices. Due to the physical constraints, it is not always possible to increase the distance ratio to a large value. Therefore, we need to know the lower threshold of the distance ratio that achieves the desired accuracy on identifying device IDs. Form our experiments, we observe that having a distance ratio above 1.3 is enough to achieve 100% accuracy in identifying device IDs. We also see that as the

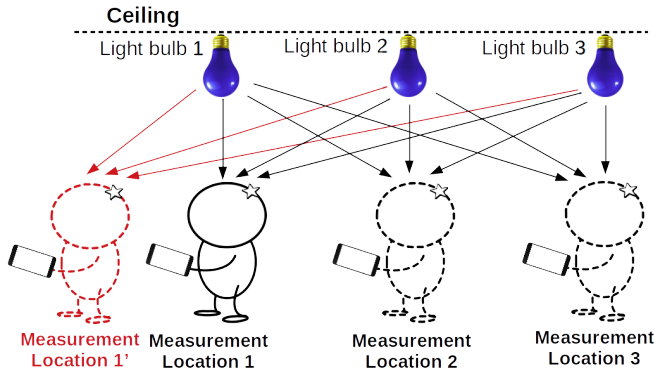


Fig. 6. Measurement procedure in MAIDE. We measure RSS at fixed positions closest to each target device. At each measurement location, we move our phone in a circular way when collecting RSS.

distance ratio gets closer to the lower bound, it may require a longer duration of measurements to achieve satisfactory accuracy. In Section 5.6, we evaluate the relationship among distance ratio, measurement duration, and accuracy. Note that, at each measurement location, we collect RSS of surrounding devices, both target and non-target, for a few seconds. After collecting the data, we derive a statistical metric, i.e., mean, median, 95 percentile (close to maximum) and 5 percentile (close to minimum)) for each device or device ID to build RSS profiles.

### 3.4 Measurement Procedure

Based on the above discussion, there are three constraints in selecting a *measurement location*. First, each measurement location corresponds to a target device, whose device ID we are interested in finding. Second, a measurement location of a target device is the position that is closest to that device compared to the other target devices. Third, a measurement location should be as close as possible to the target device. This third constraint allows us to avoid the flat-like RSS region from Figure 3(a), and to have an enough RSS contrast among multiple target devices. At each measurement location, MAIDE simultaneously collects RSS from all devices, and thus it is not time-consuming and labor-intensive.

In order to automatically select the measurement locations that meet the above three constraints, we can leverage augmented reality features (e.g., ARCore [14], ARKit [2]) of smartphones. Figure 6 depicts an example of our measurement procedure. In this example, we want to onboard three light bulbs on the ceiling. To onboard these devices, we collect RSS from all three light bulbs at three *measurement locations*. With the help of an AR session, the smartphone camera recognizes the target devices, and the location of those target devices in the AR world (i.e., the AR session). In the same AR session, the smartphone continuously monitors the position of the phone's camera. Therefore, the smartphone knows the position of the user as well as the positions of the target devices in the relative 3D coordinate system. With the position information, the AR system can guide the user to move towards a certain direction and distance to achieve the best possible distance ratio for a particular target device.

Assume that we select light bulb 1 on the ceiling in Figure 6 as our first target device. Although the measurement location 1' is closer to the target device light bulb 1 compared to other light bulbs, with the help of AR, the user will be guided to the measurement location 1, which has a higher distance ratio compared to the measurement location 1'. Once we achieve the desired distance ratio (1.3 or greater), the measurement can start automatically. In the case where

the desired distance ratio cannot be achieved due to the physical constraints, the user can manually start the measurement at the location with the highest distance ratio based on the feedback from the app. The duration for the measurement is determined based on the distance ratio, i.e., the larger the distance ratio is, the shorter the measurement duration is needed (evaluated in Section 5.6). After finishing the measurement for light bulb 1, we can select the next target device light bulb 2 to repeat the process of measurements, and so on. Once all measurements are completed for all target devices, MAIDE moves on to the mapping stage.

## 4 MAPPING PROCEDURE

For better understanding, we first formulate the problem and then describe the mapping algorithms. We further design an AR-assisted loop of measuring and mapping, to enhance system friendliness to users.

### 4.1 Problem Formulation

Assume that we have  $N$  measurement locations for  $N$  target devices. For each measurement location, we have RSS profile for  $M$  number of device IDs that include both the target and the non-target devices ( $M \geq N$ ). Correspondingly, we have an  $M$ -by- $N$  matrix  $D$ , in which  $d_{ij}$  represents the RSS profile of  $i^{th}$  ( $i = 1, 2, \dots, M$ ) device ID at  $j^{th}$  ( $j = 1, 2, \dots, N$ ) measurement location.

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1N} \\ d_{21} & d_{22} & \dots & d_{2N} \\ \dots & \dots & \dots & \dots \\ d_{M1} & d_{M2} & \dots & d_{MN} \end{bmatrix} \quad (2)$$

Given the RSS profile matrix  $D$ , our objective is to associate the right device ID  $i$  for the measurement location  $j$ .

### 4.2 Algorithms

We first describe related algorithms and then explain onboarding algorithms in MAIDE.

**4.2.1 The Naive Algorithm** [35]. For each measurement location, this algorithm selects the device ID that has the strongest RSS. The outcome of this algorithm may vary due to the different transmission powers of different devices. The algorithm complexity is  $O(M \cdot N)$ , which is polynomial.

**4.2.2 The Greedy Algorithm** [35]. This algorithm improves on the Naive Algorithm. It first finds the largest RSS in  $D$ , say RSS  $d_{ij}$ . Then it assigns measurement location  $j$  with device ID  $i$ . Afterward, the row  $i$  and column  $j$  in  $D$  is set to  $-\infty$ . The procedure repeats  $N$  times until  $N$  devices at  $N$  measurement locations are identified. Compared to the Naive Algorithm that considers a measurement location to be independent of other measurement locations, the Greedy Algorithm starts with the largest RSS (normally higher confidence) and also avoids assigning the same device ID to multiple measurement locations. The outcome of this algorithm may vary due to the different transmission powers of different devices. An optimized implementation version is to first sort the matrix into a descending array, which has a complexity of  $O(M \cdot N \cdot \log(M \cdot N))$ . Then, we traverse the array to find the first  $N$  elements with different column numbers, which has a complexity of  $O(M \cdot N)$  in the worst case. Therefore, the complexity of the Greedy Algorithm is  $O(M \cdot N \cdot \log(M \cdot N) + M \cdot N) = O(M \cdot N \cdot \log(M \cdot N))$ , which is polynomial.

**4.2.3 MAIDE Algorithms.** We propose voting-based algorithms, in which the *likelihood* of each device ID at each measurement location is considered. Equation 3 shows a vote matrix. Each device  $i$  receives a vote for location  $j$ , reflecting its likelihood of being at location  $j$ . The vote is calculated

as  $\sum_{k=1}^N (d_{ij} - d_{ik})$ . This is derived by comparing device  $i$ 's RSS at location  $j$  with other locations. A higher vote for device  $i$  at location  $j$  means that device  $i$  has greater signal strength at location  $j$  compared to that at other locations. Since each device only compares its signal strength at different locations, the vote is not affected by the difference of transmission powers between devices. Also note that the vote is jointly determined by measurement results from all locations, which makes the result more robust than the result of the Naive Algorithm and the Greedy Algorithm where a single RSS value is used. Based on the vote matrix  $V$ , we search for the largest vote summation of  $N$  elements in  $V$ . These  $N$  elements are from unique devices (i.e., different rows) and unique measurement locations (i.e., different columns).

$$V = \begin{bmatrix} \sum_{j=1}^N (d_{11} - d_{1j}) & \dots & \sum_{j=1}^N (d_{1N} - d_{1j}) \\ \sum_{j=1}^N (d_{21} - d_{2j}) & \dots & \sum_{j=1}^N (d_{2N} - d_{2j}) \\ \dots & \dots & \dots \\ \sum_{j=1}^N (d_{M1} - d_{Mj}) & \dots & \sum_{j=1}^N (d_{MN} - d_{Mj}) \end{bmatrix} \quad (3)$$

(1) The Exhaustive-MAIDE Algorithm. We propose a brute-force method [35], in which it traverses every combination of  $N$  devices out of  $M$  devices, and for those  $N$  devices, it traverses every combination of  $N$  measurement locations. The result is given by the device-location combination that has the largest summation. The algorithm requires to compare the summations of  $C_M^N \cdot N! = \frac{M!}{(M-N)!}$  combinations. Therefore, the complexity of the algorithm is  $O(\frac{M!}{(M-N)!} \cdot N)$ , which is factorial in the case of measuring at multiple locations in order to onboard devices at one run.

(2) The Greedy-MAIDE Algorithm. We propose a fast searching algorithm that only has polynomial complexity. Our algorithm has the same procedure as the Greedy Algorithm, but working with the voting matrix  $V$  rather than the RSS matrix  $D$ . Our algorithm first finds the largest voting in  $V$ , say RSS  $v_{ij}$ . Then it assigns measurement location  $j$  with device ID  $i$ . Afterwards, the row  $i$  and column  $j$  in  $V$  is set to  $-\infty$ . The procedure repeats  $N$  times until  $N$  devices at  $N$  measurement locations are identified. As with the Greedy Algorithm, this algorithm has a complexity of  $O(M \cdot N \cdot \log(M \cdot N))$ , which is polynomial. The Greedy-MAIDE Algorithm has almost the same accuracy as the Exhaustive-MAIDE Algorithm (evaluated in Section 5.4).

### 4.3 AR-Assisted Measuring and Mapping Optimization

As mentioned in Section 3.4, MAIDE leverages AR to decide the measurement locations according to our three constraints. Our measurement locations create the highest signal strength contrast among devices, and thus improve the mapping accuracy. In addition, MAIDE also exploits AR to guide users to pay more attention (e.g., taking longer measurement time) for devices that are confusing for mapping, as explained below.

Recall that our mapping algorithms output the device-location combination that has the largest summation for the likelihood vote matrix  $V$ . To locate confusing devices and locations, we compare the device-location combinations of the largest summation and the second-largest summation. If devices have the same location mapping in the largest summation and the second-largest summation combinations, they have high confidence in being correctly mapped. MAIDE removes the corresponding rows (device IDs) and columns (location IDs) in  $V$ , which results in a smaller-size  $V$  for confusing devices and locations. Then, MAIDE guides users to move to the measurement locations of the confusing devices and indicates them to measure longer at these locations. Last, MAIDE onboards the confusing devices by applying the onboarding algorithm to the smaller-size  $V$ . This process of measuring and mapping loop can be continued until the system confidently onboards all target devices.

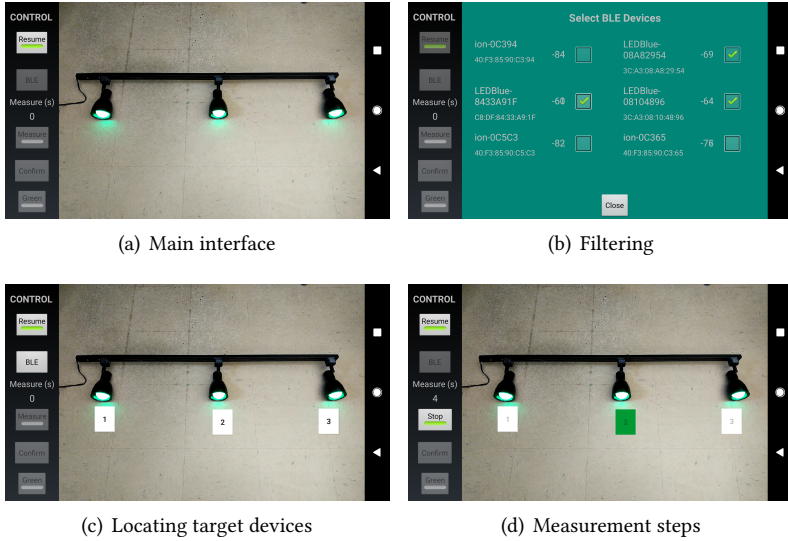


Fig. 7. Illustration of MAIDE App.

Our AR-assisted measuring-and-mapping loop not only reduces the overall measuring time (compared to the method of increasing the measurement time for all devices), but also improves the accuracy (evaluated in Section 5.9). In addition, our system is user interactive and friendly, as users receive clear and real-time guide when using our app.

#### 4.4 Clarification of Technical Contributions

MAIDE solves the three signal constraints and two physical constraints (mentioned in Section 3.1), as explained below. (1) Different transmission powers. MAIDE does not rely on the absolute RSS values of each device to onboard devices. Instead, MAIDE leverages the *change* of RSS measurements of a device at different measurement locations for device identification. (2) RSS are indistinguishable beyond a certain distance. MAIDE determines the measurement location close to the target device so that the RSS difference between different measurement locations of a device is noticeable. (3) RSS multipath effect. MAIDE proposes moving a phone in a circle way to mitigate the multipath effect. (4) Devices are not approachable. MAIDE works when the distance ratio between devices is small (i.e., they are not approachable) by carefully designing our onboarding algorithm based on the RSS comparison at multiple measurement locations. (5) Close device placement. Similar to point (4), MAIDE incorporates a voting-based algorithm that considers the likelihood of each device at each measurement location, which shows promising results even when the distance ratio between devices is small.

### 5 EVALUATION

#### 5.1 Implementation

Putting everything together, we implement MAIDE app in the Android smartphone platform. The app has about 1.6K lines of Java codes. Figure 7(a) shows the screenshot of the app, while onboarding 3 BLE light bulbs. First, the user selects the possible set of target devices based on the device names received in beacon packets (as shown in Figure 7(b)). Second, the user focuses his phone camera to the target devices. Afterward, the user clicks the light bulbs on the screen one by one. A *Label*



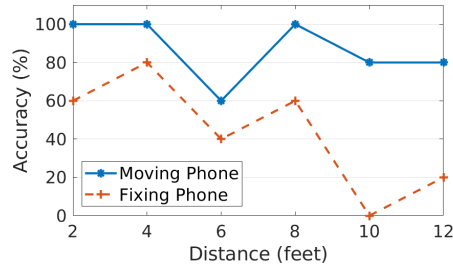


Fig. 8. Moving a phone during measurement improves the onboarding accuracy because of the mitigation of the multipath effect.

button is augmented on the screen for each light bulb (Figure 7(c)). During the measurement step, the user clicks the corresponding *Label* button on the screen for a target device (Figure 7(d)), and collects RSS reading for a few seconds at a measurement position close to the target device based on the distance ratio. The user repeats this procedure for all devices to onboard. Afterward, MAIDE automatically binds each visual object with its corresponding device ID. Then the user can control these devices, e.g., setting the brightness level of a light bulb. Note that in general, the system needs to perform two steps: (1) associate the physical device to its device ID, and (2) associate the physical device to its visual representation (e.g. image or icon) in the app. The mechanism we presented so far focuses on step 1. In a different prototype, we implement step 2, which is a smartphone AR app that recognizes the target devices [5], and their corresponding locations in an AR-session based on feature points (i.e., point clouds) extracted from ARCore SDK [14]. Then the app attaches *anchor points* to the recognized target devices. Thus it associates each physical device to its visual representation. The ARCore SDK also provides camera positions in the same AR session. After selecting the anchor point corresponding to a target device, the app guides the user to reach the measurement position and then perform a measurement for the computed duration.

## 5.2 Experiment Setup

We use Iotton BLE beacons and MagicLight BLE light bulbs as transmitters, and different smartphone models such as Huawei Mate 20 and Google Pixel 2 as receivers. We mainly deploy BLE devices at three sites: a small meeting room, a medium conference room, and an office corridor at Bell Labs, Holmdel, New Jersey. The selected sites are representative for enterprise environments.

**Data Collection.** We deploy BLE devices on ceilings and make them into a line topology and a grid topology, and collect RSS data at each measurement location for 30 seconds. The distance between the phone and the ceiling is about 6 feet. For the line topology, we put 4 devices in the small meeting room, the medium conference room, and the office corridor, and make them into a line topology with 2 feet apart between adjacent devices. In total, we collect 33 traces for the line topology. For the grid topology, we put 6 devices into a 2-by-3 grid in the medium conference room with 4 feet apart between adjacent devices. In total, we collect 15 traces for the grid topology. We also collect 10 traces for devices on the ceiling of the office corridor with various distances between them (i.e., 6 feet, 8 feet, and 10 feet).

## 5.3 Mitigating Multipath Effect with Moving Phones

We first evaluate whether doing a circular movement while holding the phone during measurement can reduce the multipath effect and improve accuracy. This is done by comparing circular movement with the case holding phone static at a fixed location. Here we put two devices on the floor instead



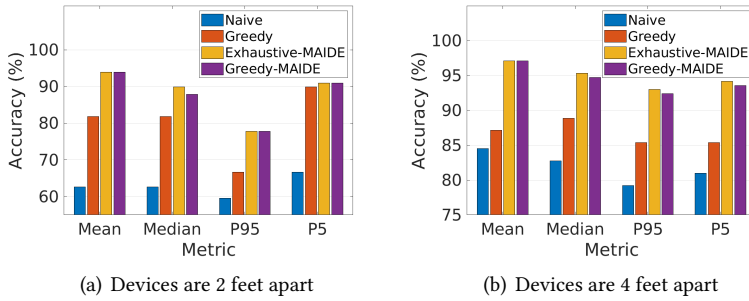


Fig. 9. Accuracy of onboarding two devices on ceilings with different onboarding algorithms.

Algorithm	Topology: Line	Topology: Grid
	(2 feet apart on ceiling)	(4 feet apart on ceiling)
Naive	53.8% (median)	62.2% (mean)
Greedy	76.5% (mean)	64.4% (median)
MAIDE	87.9%	84.4%

Table 1. Accuracy of onboarding multiple devices that are shaped into a line topology and a grid topology.

of on ceilings for the convenience of experimenting with a wider range of distances. The devices are placed 4 feet apart, one left and one right. Then we hold the phone 2 feet away from the devices, measure once at the left side and once at the right side, each time collecting 30 seconds’ data. This process is repeated for the two measurement approaches at different measurement distances up to 12 feet. Then we use the Greedy Algorithm to onboard these devices.

Figure 8 shows the onboarding accuracy of moving a phone and fixing a phone during measurements. As we can see, moving the phone during measurements greatly improves the onboarding accuracy. This is because RSS data are averaged in a region to combat the multipath effect. Therefore, we always apply circular movements for the rest experiments.

#### 5.4 Comparison among Onboarding Algorithms

We investigate the accuracy of different onboarding algorithms including Naive, Greedy, Exhaustive-MAIDE, and Greedy-MAIDE. We use the collected data of two devices with distance of either 2 feet or 4 feet in between. Therefore, the maximum difference of distances between the pair of devices and the phone is only 0.3 feet (for the 2-feet case) and 1.2 feet (for the 4-feet case), respectively.

Figure 9 shows the overall accuracy comparison among these algorithms. It clearly shows that the voting-based algorithms (i.e., exhaustive-MAIDE and greedy-MAIDE) outperform the Greedy algorithm which in turn outperforms the Naive algorithm. The Greedy-MAIDE achieves similar accuracy as the Exhaustive-MAIDE, both reaching 93.4% and 97.1% accuracy using the Mean metric for 2 feet and 4 feet device distances, respectively. In the rest of the evaluation, we use the Exhaustive-MAIDE algorithm and simply call it the MAIDE algorithm.

#### 5.5 Devices in Different Topology

Table 1 tabulates the accuracy for the line topology and the grid topology. We use Mean in our algorithm, and compare to the other algorithms with whichever metric gives its highest accuracy. As we can see, MAIDE achieves higher accuracy than the Greedy algorithm, which in turn is more

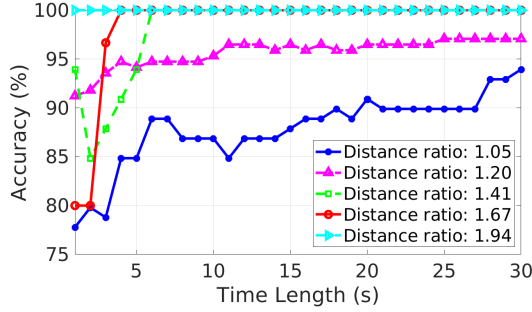


Fig. 10. Accuracy of onboarding 2 devices with different distance ratios versus measurement time length.

Distance Ratio	1.05	1.20	1.41	1.67	1.94
Measurement Cutoff time	>	>	6s	4s	1s

Table 2. Measurement cutoff time for different distance ratios that guarantee 100% accuracy. > means that longer measurement time is preferred.

accurate than the Naive algorithm. More specifically, MAIDE achieves an average 86.2% accuracy for these two topologies, with an average improvement of 15.7% and 28.2% compared to the Greedy algorithm and the Naive algorithm, respectively.

### 5.6 Measurement Duration Versus Distance Ratio

We study the measurement cutoff duration for a given distance ratio that guarantees 100% onboarding accuracy. We use the collected data for 2 devices that are 2, 4, 6, 8, and 10 feet apart. We measure at 6 feet below, and thus the corresponding distance ratios are 1.05, 1.20, 1.41, 1.67, and 1.94, respectively.

Figure 10 shows the accuracy of onboarding 2 devices with different distance ratios versus measurement time length. Generally, longer measurement time gives higher accuracy. On the other hand, it is also clear that for larger device distance ratios shorter measuring time is enough to guarantee 100% accuracy. This is because a larger distance ratio leads to larger signal level contrast and hence easier to distinguish. Table 2 tabulates the measurement cutoff time that guarantees 100% accuracy. For distance ratios that are too small which cannot provide 100% accuracy, we use the symbol > to indicate that longer measurement time is preferred in order to have higher onboarding accuracy. The results show that a decreasing measurement time is required for larger distance ratios to guarantee 100% accuracy. For example, we only need to measure 1 second's data with 100% accuracy when the distance ratio is 1.94.

### 5.7 Interference of Non-target Devices

We explore how non-target devices affect our onboarding algorithm. This is the case where the number of measurement locations is smaller than the total number of devices since we only measure at target device locations. We put two devices on a ceiling as target devices with 2 feet apart, and a third device at one side as a non-target device, along the line of the target devices. We regard the non-target device as the near interferer when it is 2 feet away from the closet target device and as the far interferer when it is 4 feet away from the closet target device.

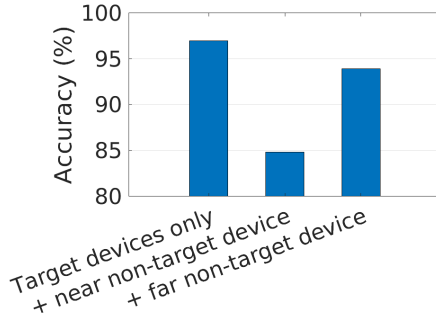


Fig. 11. Interference of non-target devices with different distances to the target device.

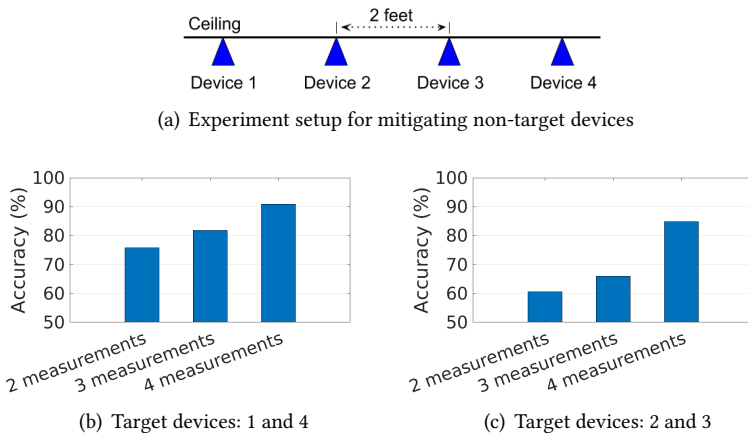


Fig. 12. Mitigating interference of non-target devices by making more measurements at locations other than target devices’.

Figure 11 shows the onboarding accuracy with interferers. The first bar represents the baseline case of onboarding target devices without interferers. The next two bars represent the onboarding accuracy of target devices with either near or far interferer, respectively. As expected, the near interferer degrades the onboarding accuracy more significantly than the far interferer.

### 5.8 Mitigating Interference of Non-Target Devices

To mitigate the interference of non-target devices, we conduct experiments to explore whether making more measurements at locations other than target devices’ can improve onboarding accuracy. We again use the line topology shown in Figure 12(a). We consider the following two cases: (1) Device 1 and device 4 are target devices, and device 2 and device 3 are interferers. In this case, we randomly pick location 2 or location 3 when making one extra measurement, and pick both location 2 and location 3 when making two extra measurements. (2) Device 2 and device 3 are target devices, and device 1 and device 4 are interferers. In this case, we randomly pick location 1 or location 4 when making one extra measurement, and pick both location 1 and location 4 when making two extra measurements.

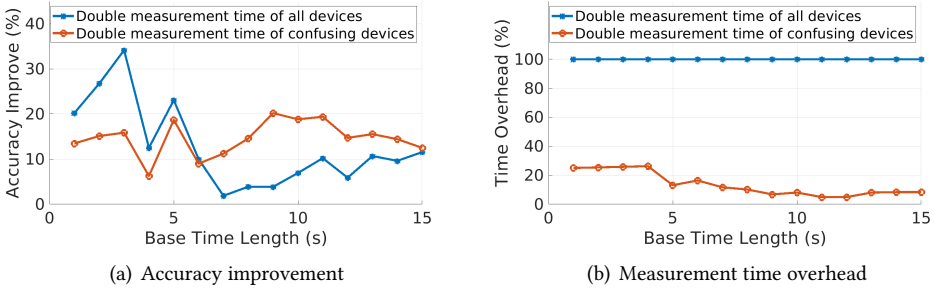


Fig. 13. AR-assisted measuring and mapping optimization, by taking longer measurement time for confusing devices.

Figure 12 shows the onboarding accuracy versus the number of measurements. The first bar represents the accuracy of onboarding two target devices out of four devices by only measuring at the target devices (i.e., two measurements). The next two bars represent making 3 and 4 measurements, respectively. The results show that making more measurements consistently improves onboarding accuracy. We can see 15.2% improvement and 24.2% improvement for case 1 and case 2 respectively when making two more measurements. Hence a natural conclusion is that in the cases where non-target devices exist close to the target devices, we should make additional measurements at the non-target device locations to improve onboarding accuracy.

### 5.9 AR-Assisted Measuring and Mapping Optimization

We investigate the benefits of the AR-assisted measuring and mapping procedures. In addition to deciding the measurement locations, AR determines confusing devices from the mapping result, and guides users to measure longer for these devices. We use the collected data for 2 devices that are 2 feet apart on ceiling. The measurement time length is from 1 to 15 seconds (referred as base time length in this experiment). We double the measuring time for confusing devices, and compare with the case when we double the measurement time for all devices.

Figure 13(a) compares the accuracy of our AR-assisted method and the method of doubling all devices' measuring time. Both methods have higher accuracy than the baseline, since longer measurement time results in higher accuracy. It is interesting to note that, from 6 seconds onward, our method of doubling measurements of only confusing devices achieves even higher accuracy than doubling all devices' measurement time. This is probably because once we remove confident device-location mappings from the vote matrix  $V$ , the problem size of remaining confusing devices and locations is simpler and less uncertain. Figure 13(b) shows the extra measurement overhead of both methods, comparing to the baseline. In our AR-assisted method, we only need to take extra measurement for confusing devices, which is few in general. Therefore, our overhead is low, e.g., less than 10% measurement time for 7 seconds onward.

### 5.10 Approachable Scenario

In an approachable setting, where we deploy 5 devices at various positions such as on table, floor and TV top in the medium conference room. In this case, MAIDE achieves 100% accuracy. Here the result is as expected, as we have measured RSS very close to the target device. Thus the likelihood of that device at its measurement location is significantly higher than at other measurement locations. Our previous evaluations focus on more challenging scenarios when devices are not approachable.

## 6 DISCUSSION

### 6.1 Application Scenarios

MAIDE has many potential applications, including large-scale device onboarding for industry automation, inventory management, and interactive indoor maps.

*Industry Automation.* Industries are adopting machines that are equipped with wireless sensors and actuators for large-scale automation. Operating these sensors and actuators remotely is the cornerstone of industrial automation. This requires registering them in the automation system. In order to do so, the current de-facto process is to either enter each sensor/actuator ID into the system manually, if this can be found in the original package; or through a trial-and-error process where the operator can try to connect to the device and change its status (e.g. turn them on/off or change status), and then observe the change and accordingly make the association. However, such a manual process may be error-prone and inefficient. Instead, the proposed MAIDE mechanism can simply use a phone to do the signal feature measurement close to each installed gadget. After all the measurements are made, MAIDE automatically associates all gadgets' IDs with their corresponding locations.

*Interactive Indoor Map for Enterprise.* As enterprise buildings are continuously installing more and more smart devices and appliances, provisioning and interacting with them is becoming a challenge. One convenient way for the users to interact with these devices is to use a smartphone app with an interactive indoor map of the building [7], where the smart devices are marked on the map. With this, users can click on the devices on the map to control them. At the same time, admin users can provision who is allowed to control which device at what time. In order to support such an app, one needs to create a database of device IDs and their corresponding locations in the indoor map. We need a streamlined process to create such a database dynamically for any indoor environments whenever new devices are installed or replaced. We believe, in this usage scenario, MAIDE can help in assigning device IDs on the map by associating the physical device to its beacon and MAC address.

*Inventory Management in Warehouse.* In order to maintain a large inventory of IoT devices of the same type (from the same manufacturer) in real-time, one has to keep a record of the device IDs. For example, consider a hospital environment where patients are allocated smart heart-rate monitors throughout their stay in the hospital. The devices are reassigned to new patients as old ones are being charged. The de-facto process requires to first register all devices in the inventory manually by entering their MAC addresses and serial numbers etc. into a database, and also attach a printed label with its unique ID to this device. Then when the device leaves the inventory, a scanner could be used to update the record. However, such a manual process may be error-prone and inefficient. Instead, our onboarding solution can be used to manage the day-to-day inventory. For instance, we can simply hold a smartphone close to the IoT device, and MAIDE automatically identifies the device based on its beacon signal, despite having other beacon signals from similar nearby devices.

### 6.2 Practicability of MAIDE

Our evaluation settings are representative and thus we believe MAIDE can be readily applied to other environments. We conduct experiments in three typical enterprise sites: a small meeting room, a medium conference room, and an office corridor. As MAIDE can easily onboard devices that are approachable (Section 5.10), we target challenging scenarios where devices are installed on the ceiling. We collect more than 10 traces at each site during office hours when people are present nearby.

Although RSS profiles are subject to environmental changes, our data collection method of moving a phone during measurements mitigates the effect. During our data collection in the

Related Area	Publication	Category	Limitation for IoT Onboarding
<b>Device Localization</b>	[1, 16, 21]	Camera-based	Infrastructure and privacy concern
	[18, 19]	VLC-based	Infrastructure, light flicker, and availability
	[3, 17, 27]	RF-based	Infrastructure and sensitivity
<b>RSS Multipath Mitigation</b>	[20, 28]	Fingerprinting	Laboring and time-consuming
	[15, 36]	Propagation Modeling	Detailed hardware information
<b>Device Identification</b>	[10, 34]	Hardware-based	Only work for specified devices
	[22, 23, 25]	Trajectory-based	Cannot work for fixed devices

Table 3. Comparison of representative related works.

building, people occasionally walked nearby, but our system still shows promising performance. Nevertheless, there are several problems are worth to explore. (1) Detailed analysis of under what circumstances our system wrongly onboard devices. For example, the grid topology has lower accuracy than the line topology (Table 1). Since we need to onboard 6 devices in the grid and only 4 devices in the line, it may be prone to jump into the conclusion that our onboarding algorithm does not scale with the number of devices. However, we conducted an experiment with 10 devices on a ceiling and did not observe a clear pattern of accuracy change with the number of devices to onboard. (2) The accuracy of some of our onboarding results is not very high. This is because we target challenging device setups. Most of our experiments are done with devices that are only 2 feet apart on a ceiling, which corresponds to a distance ratio of 1.05 from these devices to the phone. In other words, these devices have approximately the same distances to the phone. For decent distance ratios, e.g., larger than 1.4, MAIDE can easily separate these devices (Figure 10). Our current conjecture is that a distance ratio larger than 1.3 in a typical enterprise and industry environment similar to those in our experiments can give 100% accuracy. Although the thresholds of distance ratio may not be the same in different environments, the principle of identifying measurement locations remains valid, and a longer measurement time can be adopted to guarantee onboarding accuracy. Nonetheless, more experiments are still needed to verify and quantify.

In general, MAIDE is not affected by the transmission powers of IoT devices because it does not directly use absolute RSS values. Instead, it is based on the difference of RSS at different measurement locations. The presence of WiFi devices may potentially affect the measurement for BLE devices due to channel overlap. However, during our experiment, the inference of WiFi signals does not seem to have much impact.

## 7 RELATED WORK

In this paper, we target a unique problem of IoT onboarding in a practical way. A practical IoT onboarding system needs to work with any kind of wireless devices without relying on extra/impractical information such as device locations (commonly required by tracking systems), device hardware details (e.g., transmission powers, and receiver gains for channel modeling), and offline environmental fingerprinting. To the best of our knowledge, no existing work can be applied directly in this challenging application setting. Therefore, in Section 5, we evaluate MAIDE without comparing it with other systems. Nonetheless, some works are related to the components of MAIDE. We tabulate representative related works in Table 3 regarding device localization, RSS multipath mitigation, and device identification.

*Device Localization.* A localization system can be used to locate the user or her mobile device and associate the device ID of the strongest RSS with the closest IoT device. A plethora of localization systems have been proposed, which can be roughly classified based on the data sources: (1) Camera-based. A number of cameras can be deployed on-site. With known camera locations, an object

can be localized through the geometry relationship (e.g., triangulation [16]) of the object in each camera's view. However, it is costly to build a camera-based localization system. An alternative solution is to use Simultaneous Localization and Mapping (SLAM) [21] to online track the user's device. However, SLAM systems are heavy for mobile devices and thus require edge servers for offloading [1]. In addition to the infrastructure cost, camera-based localization systems have severe privacy issues. (2) Visible Light Communication (VLC)-based. Specialized LEDs can be used to build localization systems such as Epsilon [19] and Luxapose [18]. Compared to camera-based systems, VLC-based systems mitigate privacy issues. However, without careful system designs, they could cause light flickers, which is harmful to humans [29]. In addition, VLC-based systems can only work when the LEDs are turned on, and thus, they are not always available. (3) RF-based. RF-based techniques have been proposed to solve localization problems [3, 17, 27]. For example, DLoc [3] models the indoor WiFi environment by finger-printing and then localizes the target device in the generated map. However, RF-based localization systems are sensitive to environmental changes. As a result, they could lead to low accuracy in complex indoor environments. Compared to the localization systems, MAIDE is an infrastructure-free solution.

*RSS Multipath Mitigation.* RSS is severely affected by the multipath effect, resulting in extremely low localization accuracy when a proper multipath mitigation process is not incorporated. Prior research has shown that using only RSS for localization has a median estimation error greater than five meters [30], which makes it challenging to distinguish devices that are less than 1 meter apart. There are two main methods to mitigate RSS multipath effect. (1) Fingerprinting. Environmental constraints and fingerprinting can be used to establish the relationship between RSS measurements and locations [20, 28]. However, finger-printing is laboring and time consuming for tasks such as IoT onboarding. (2) Propagation modeling. Many path propagation models have been designed to mitigate RSS multipath in indoor environments [15, 36]. They achieve higher ranging accuracy than the classical log-distance path loss model. However, path propagation models require detailed hardware information such as transmission power (dependent on the battery level) and antenna gains of senders and receivers, which are inconvenient to obtain in practice. Compared with these methods of mitigating the RSS multipath effect, MAIDE moves a phone in a circle way while making measurements, which shows promising results.

*Device Identification.* Some works have studied the problem of identifying sources (devices) from the received data/signal. They can be roughly divided into two categories. (1) Hardware-based. The subtle difference of hardware has been leveraged to identify devices. For example, cameras can be identified based on the captured images [10]; fluorescent lights and conventional LEDs can be identified by leveraging the intrinsic optical emission features of these light devices [34]. However, these works target one specific device, e.g., LEDs and cameras, which cannot apply to IoT devices of different varieties. (2) Trajectory-based. In this kind of work, the devices are identified by matching the movement trajectory of devices with the change of sensor or wireless signals. For example, IDrone [25] identifies drones by matching the drones' trajectories from the external cameras and the embedded inertial sensors. Similarly, UniverSense [22] identifies an IoT device by moving the device in front of a camera and matches the device's trajectory and inertial sensor readings. In addition to the inertial sensors, wireless signals have also been exploited in device identification. VisIoT [23] visualizes the source of the wireless signal on the screen with specialized and customized hardware of directional antenna arrays. However, trajectory-based device identification methods have limitations. Most IoT devices are fixed in place and cannot be easily moved, e.g., those already deployed on ceilings. Also, it is expensive to build a vision or specialized identification system for IoT applications. In comparison, MAIDE is a lightweight solution that works for any wireless IoT device.

## 8 CONCLUSION

In this paper, we propose MAIDE that targets an emerging necessity to onboard IoT devices in an intuitive and convenient way. MAIDE is a streamlined system that consists of three stages to onboard devices: filtering, measurement, and mapping. In the filtering stage, we collect broadcast advertisement messages from surrounding devices. From these beacon packets, we select the possible set of devices whose identities are potential candidates for the target devices. In the measurement stage, we measure the signal features of the received advertisement messages from the selected candidate devices at a unique measurement location for each target device. In the mapping stage, we use a novel voting-based algorithm based on the measurement at different locations to infer the identities of the target devices. MAIDE does not require modifications to IoT devices and infrastructure support. It can run on COTS smartphones as an app, and thus it is readily available. In addition, our evaluation results show that MAIDE achieves high accuracy with short measurement time.

## REFERENCES

- [1] Ali J. Ben Ali, Zakieh Sadat Hashemifar, and Karthik Dantu. 2020. Edge-SLAM: Edge-Assisted Visual Simultaneous Localization and Mapping. In *International Conference on Mobile Systems, Applications, and Services (MobiSys)*. 325–337.
- [2] Apple Developer. [n. d.]. ARKit Augmented Reality. <https://developer.apple.com/augmented-reality/>. Accessed on September 19, 2019.
- [3] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. 2020. Deep Learning based Wireless Localization for Indoor Navigation. In *Proceedings of the 26th annual international conference on Mobile computing and networking (MobiCom)*. 1–14.
- [4] Justin Chan, Anran Wang, Vikram Iyer, and Shyamnath Gollakota. 2018. Surface MIMO: Using Conductive Surfaces For MIMO Between Small Devices. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom)*. 3–18.
- [5] Jongwon Choi, Hyung Jin Chang, Tobias Fischer, Sangdoo Yun, Kyuewang Lee, Jiyeoup Jeong, Yiannis Demiris, and Jin Young Choi. 2018. Context-aware Deep Feature Compression for High-speed Visual Tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 479–488.
- [6] Mario Colotta, Giovanni Pau, Timothy Talty, and Ozan K. Tonguz. 2018. Bluetooth 5: A Concrete Step Forward toward the IoT. *IEEE Communications Magazine* (2018), 125–131.
- [7] CUBICASA. [n. d.]. Integrate Floor Plans into Your Smart Home and Home Automation Applications. <https://www.cubi.casa/floor-plans-smart-homes-iot/>. Accessed on September 19, 2019.
- [8] Ivan Cvitic, Dragan Perakovic, Marko Perisa, and Brij Gupta. 2021. Ensemble Machine Learning Approach for Classification of IoT Devices in Smart Home. *International Journal of Machine Learning and Cybernetics* (2021), 1–24.
- [9] Vinko Erceg, Larry J. Greenstein, Sony Y. Tjandra, Seth R. Parkoff, Ajay Gupta, Boris Kulic, Arthur A. Julius, and Renee Bianchi. 1999. An Empirically Based Path Loss Model for Wireless Channels in Suburban Environments. *IEEE Journal on Selected Areas in Communications* 17, 7 (1999), 1205–1211.
- [10] David Freire-Oregon, Fabio Narducci, Silvio Barra, and Modesto Castrillon-Santana. 2018. Deep Learning for Source Camera Identification on Mobile Devices. *Pattern Recognition Letters* (2018), 1–6.
- [11] Jennifer Gilburg. 2017. Zero Touch Device Onboarding for IoT Control Platforms. RSA Conference.
- [12] Jon Gjengset, Jie Xiong, Graeme McPhillips, and Kyle Jamieson. 2014. Phaser: Enabling Phased Array Signal Processing on Commodity WiFi Access Points. In *Proceedings of the 20th annual international conference on Mobile computing and networking (MobiCom)*. 153–164.
- [13] Andrea Goldsmith. 2005. *Wireless Communications*. Cambridge University Press.
- [14] Google Developers. [n. d.]. ARCore Augmented Reality. <https://developers.google.com/ar/>. Accessed on September 19, 2019.
- [15] Yao Guo, Kaide Huang, Nanyong Jiang, Xuemei Guo, Youfu Li, and Guoli Wang. 2015. An Exponential-Rayleigh Model for RSS-Based Device-Free Localization and Tracking. *IEEE Transactions on Mobile Computing (TMC)* 14, 3 (2015), 484–494.
- [16] Jeremie Housineau, Daniel E. Clark, Spela Ivekovic, Chee Sing Lee, and Jose Franco. 2016. A Unified Approach for Multi-Object Triangulation, Tracking and Camera Calibration. *IEEE Transactions on Signal Processing* 64, 11 (2016), 2934–2948.
- [17] Swaran Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. 2014. Accurate Indoor Localization with Zero Start-up Cost. In *Proceedings of the 20th annual international conference on Mobile computing and networking (MobiCom)*.



MAIDE: Augmented Reality (AR)-facilitated Mobile System for Onboarding of Internet of Things (IoT) Devices at Ease  
483–494.

- [18] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. 2014. Luxapose: Indoor Positioning with Mobile Phones and Visible Light. In *International Conference on Mobile Computing and Networking (MobiCom)*. 447–458.
- [19] Liqun Li, Pan Hu, Chunyi Peng, Guobin Shen, and Feng Zhao. 2014. Epsilon: A Visible Light based Positioning System. In *USENIX Conference on Networked Systems Design and Implementation (NSDI)*. 331–343.
- [20] Yan Li, Simon Williams, Bill Moran, Allison Kealy, and Guenther Retscher. 2018. High-Dimensional Probabilistic Fingerprinting in Wireless Sensor Networks Based on a Multivariate Gaussian Mixture Model. *Sensors* 18, 8 (2018), 1–24.
- [21] Raul Mur-Artal and Juan D. Tardos. 2017. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics* 33, 5 (2017), 1255–1262.
- [22] Shijia Pan, Carlos Ruiz, Jun Han, Adeola Bannis, Patrick Tague, Hae Young Noh, and Pei Zhang. 2018. UniverSense: IoT Device Pairing through Heterogeneous Sensing Signals. In *Proceedings of the 19th International Workshop on Mobile Computing Systems and Applications (HotMobile)*. 55–60.
- [23] Yongtae Park, Sangki Yun, and Kyu-Han Kim. 2019. When IoT met Augmented Reality: Visualizing the Source of the Wireless Signal in AR View. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*.
- [24] Eric James Rongo and Marcello Vincenzo Liroy. 2015. Peer-to-Peer Onboarding of Internet of Things (IOT) Devices over Various Communication Interfaces. US Patent: US20150121470A1.
- [25] Carlos Ruiz, Shijia Pan, Adeola Bannis, Xinlei Chen, Carlee Joe-Wong, Hae Young Noh, and Pei Zhang. 2018. IDrone: Robust Identification through Motion Actuation Feedback. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 2, 2 (2018), 1–22.
- [26] Abdel Mlak Said, Aymen Yahyaoui, and Takoua Abdellatif. 2021. Efficient Anomaly Detection for Smart Hospital IoT Systems. *Sensors* 12, 4 (2021), 1–24.
- [27] Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. 2013. Avoiding Multipath to Revive Inbuilding WiFi Localization. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services (MobiSys)*. 249–262.
- [28] Shuai Sun, Xuezhi Wang, Bill Moran, Akram Al-Hourani, and Wayne S.T. Rowe. 2019. Radio Source Localization Using Received Signal Strength in a Multipath Environment. In *International Conference on Information Fusion (FUSION)*. 1–6.
- [29] T Tashiro, S Kawanobe, T Kimura-Minoda, S Kohko, T Ishikawa, and M Ayama. 2015. Discomfort Glare for White LED Light Sources with Different Spatial Arrangements. *Lighting Research and Technology* 47, 3 (2015), 316–337.
- [30] Daniel Turner, Stefan Savage, and Alex C. Snoeren. 2011. On the Empirical Performance of Self-calibrating WiFi Location Systems. In *IEEE 36th Conference on Local Computer Networks (LCN)*. 76–84.
- [31] Xiufeng Xie, Kang G. Shin, Hamed Yousefi, and Suining He. 2018. Wireless CSI-based Head Tracking in the Driver Seat. In *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*. 112–125.
- [32] Jie Xiong, Karthikeyan Sundaresan, and Kyle Jamieson. 2015. ToneTrack: Leveraging Frequency-Agile Radios for Time-Based Indoor Wireless Localization. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom)*. 537–549.
- [33] Li Da Xu, Wu He, and Shancang Li. 2014. Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics* 10, 4 (2014), 2233–2243.
- [34] Chi Zhang and Xinyu Zhang. 2017. Pulsar: Towards Ubiquitous Visible Light Localization. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom)*. 208–221.
- [35] Huanle Zhang, Mostafa Uddin, Fang Hao, Sarit Mukherjee, and Prasant Mohapatra. 2019. AIDE: Augmented Onboarding of IoT Devices at Ease. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (HotMobile)*. 123–128.
- [36] Yu Zhao, Yunhuai Liu, Tingting Yu, Tian He, and Chen Qian. 2018. FREDI: Robust RSS-based Ranging with Multipath Effect and Radio Interference. *Computer Networks* 147 (2018), 49–63.