

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/327757136>

CrowdProbe: Non-invasive Crowd Monitoring with Wi-Fi Probe

Article · September 2018

DOI: 10.1145/3264925

CITATION

1

READS

312

3 authors, including:



Hande Hong

National University of Singapore

14 PUBLICATIONS 108 CITATIONS

SEE PROFILE

CrowdProbe: Non-invasive Crowd Monitoring with WiFi Probe

HANDE HONG*, National University of Singapore, Singapore

GIRISHA DURREL DE SILVA, National University of Singapore, Singapore

MUN CHOON CHAN, National University of Singapore, Singapore

Devices with integrated Wi-Fi chips broadcast beacons for network connection management purposes. Such information can be captured with inexpensive monitors and used to extract user behavior. To understand the behavior of visitors, we deployed our passive monitoring system—CrowdProbe, in a multi-floor museum for six months. We used a Hidden Markov Models (HMM) based trajectory inference algorithm to infer crowd movement using more than 1.7 million opportunistically obtained probe request frames.

However, as more devices adopt schemes to randomize their MAC addresses in the passive probe session to protect user privacy, it becomes more difficult to track crowd and understand their behavior. In this paper, we try to make use of historical transition probability to reason about the movement of those randomized devices with spatial and temporal constraints. With CrowdProbe, we are able to achieve sufficient accuracy to understand the movement of visitors carrying devices with randomized MAC addresses.

CCS Concepts: • Networks → Location based services; • Human-centered computing → Mobile phones; • Mathematics of computing → Kalman filters and hidden Markov models;

Additional Key Words and Phrases: Passive tracking, randomization, transition probability, Crowd movement

ACM Reference Format:

Hande Hong, Girisha Durrel De Silva, and Mun Choon Chan. 2018. CrowdProbe: Non-invasive Crowd Monitoring with WiFi Probe. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 115 (September 2018), 23 pages. <https://doi.org/10.1145/3264925>

1 INTRODUCTION

Understanding how crowds move and how they behave has been one of the focuses for the research community. Gaining such information is of vital importance for managing visitor flow in public areas such as shopping malls, railway stations, and museums. By knowing how people move, we are able to come up with countermeasures to reduce congestion and improve the spatial arrangement. Furthermore, we can foresee a visitor's future movement based on statistical patterns.

The most traditional way of tracking is to use pencil and paper to record how users move along with the corresponding timestamps. Such a method is labor-intensive and tedious. It is also error-prone when there is a large crowd. The ubiquity of digital devices and technologies have revolutionized the way we get to know about our environment. Video-based recognition is one of the most popular technologies used to observe visitor

*This is the corresponding author

Authors' addresses: Hande Hong, National University of Singapore, Singapore, honghand@comp.nus.edu.sg; Girisha Durrel De Silva, National University of Singapore, Singapore, girisha@comp.nus.edu.sg; Mun Choon Chan, National University of Singapore, Singapore, chanmc@comp.nus.edu.sg.

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.

© 2018 Association for Computing Machinery.

2474-9567/2018/9-ART115 \$15.00

<https://doi.org/10.1145/3264925>

behavior[4, 22, 24–26]. However, the deployment of the video-based system is expensive and the system could potentially have poor performance because of limited lighting condition and overlapped individuals in the same image. Furthermore, people are concerned about privacy and not willing to be subjected to visual monitoring. To overcome the above limitations, researchers have looked to exploit different technologies including the use of Bluetooth[19, 32], cellular network[2] and, RFID[6, 33].

Due to the widespread deployment of WiFi networks and the availability of WiFi chipsets on smartphones, use of WiFi related information to extract user information has been both popular and shown to be effective[1, 3, 5, 7]. Smartphones periodically broadcast probe request frames to trigger responses from nearby APs. By deploying WiFi monitors in the environment, we can capture these management frames and extract location information related to phone owners. Such methods are *passive* because they require no change on the mobile devices. Passive scanning is performed only by the WiFi monitors with no impact on the operations of existing infrastructure. While previous work[15, 23] has shown the feasibility of such methods, iOS and Android have enabled MAC randomization to protect user privacy. This adds to the challenge of whether such technique can be used in practice.

In this paper, we present CrowdProbe, a system that has been deployed in a multi-floor museum to track thousands of visitors daily using passive WiFi monitoring over six months. We input temporally and spatially sparse passively collected RSS fingerprints to a Hidden Markov Models(HMM) based model to generate visitor trajectories. Different from traditional HMM, we do not obtain regular observations from the system since the probe requests are only sent opportunistically and can be quite sparse. Instead, we modify the model to include specific features of museum visitors to improve the trajectory inference performance. In addition, we make use of historical transition probability to reason about the movement of those randomized devices with spatial and temporal constraints. We summarize our contributions as follows:

- To the best of our knowledge, CrowdProbe is the first large-scale passive WiFi monitoring system deployed in a complex indoor public space. Six months' experience and data we get can be valuable in bridging research and practical usage.
- We use Hidden Markov Models(HMM) based trajectories generation method which makes use of WiFi fingerprinting, spatial constraints and temporal constraints. With the proposed method, we successfully generate more than 91 thousand traces which give adequate information to understand visitor behavior in the museum.
- Based on the data accumulated in the visitor traces, we generate visitor transition probability and show that this information can be used to accurately reason about the short time crowd movement of the visitors with mobile devices with randomized MAC address.

The rest of the paper is organized as follows. We give the background of probe request and MAC randomization in Section 2. In Section 3, we describe the architecture for the CrowdProbe system and deployment setting. In Section 4, we present how the data is processed for trajectory inference. We present our trajectory inference algorithm in Section 5. We use the transition probability generated by the trajectory to infer the movement of the visitors with mobile devices with randomized MAC address in Section 6. The evaluation of CrowdProbe is given in Section 7. Then we present the related work and discussion in Section 8 and Section 9. Finally, we summarize the paper in Section 10.

2 BACKGROUND

2.1 Probe Request

Smartphone broadcasts probe request frames to trigger responses from nearby APs with the purpose of speeding up the discovery of surrounding APs. Such frames are management frames containing information such as network identifier (SSID), MAC address, signal strength, and the time stamp. The emission of such a frame is

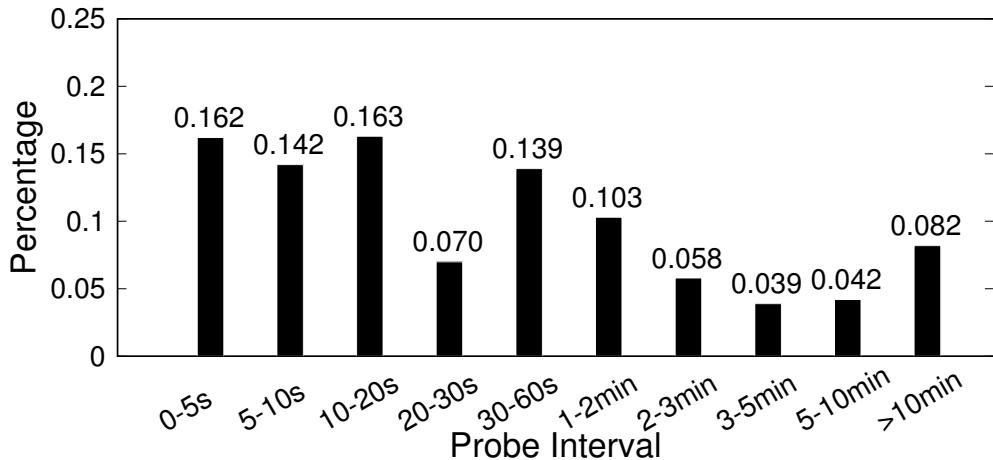


Fig. 1. Probe request interval distribution from data collected in museum

unavoidable as long as the device needs to connect to the network. Devices generally send probe request frames when they are not associated. However, when the currently connected WiFi signal becomes weak, the device will start to send probe frames to find better network candidate and prepare for handover. Such features make it suitable for indoor tracking as most of the indoor environments have complex layouts. When a visitor moves around inside the building, WiFi signal can vary a lot and trigger another probe to be sent from the mobile device.

To understand how frequently probe requests are sent in real life scenarios, we process the data we collect in the museum and plot the result in the Figure 1. As can be seen from the figure, probe request frames can be sent with intervals ranging from 5 seconds to more than 10 min, with 88% of the frames sent within 5 min. In places like shopping malls, museums, and other public spaces, visitors can spend up to an hour or more. The information provided by the probe requests can provide up to minute-level granularity on coarse user locations and thus can help us understand the movement of visitors in these public spaces.

2.2 MAC Randomization

2.2.1 iOS. From iOS 8 onward, Apple introduced MAC address randomization to avoid passive tracking of devices. The initial setting is that randomized addresses are used only while the devices are not associated and in sleep mode[13]. In later versions, the condition to trigger randomization has been extended to include location service and auto-join scan [29]. This means that devices are sending more randomized MAC address in the probe frame. From previous work in [21], we know that Apple device seems to implement true randomization across the entire field of MAC address.

2.2.2 Android. Following the same pace as iOS, Google's Android operating system added experimental support for MAC randomization. Full implementation went live in version 6.0 which covers most of the Android user base. However, a recent study shows that Android's MAC randomization is largely absent[14] even if the OS version does support this feature. Compared to Apple device, Android devices, for example, Google devices are always randomized with prefix DA:A1:19.

2.2.3 MAC Randomization Implementation in Practice. We made an analysis of the museum data regarding MAC randomization and show the statistics in Table 1. Among all the probe request frames we have collected, 63% of the probe request frames were sent with randomized MAC addresses. If devices have similar probe frequency,

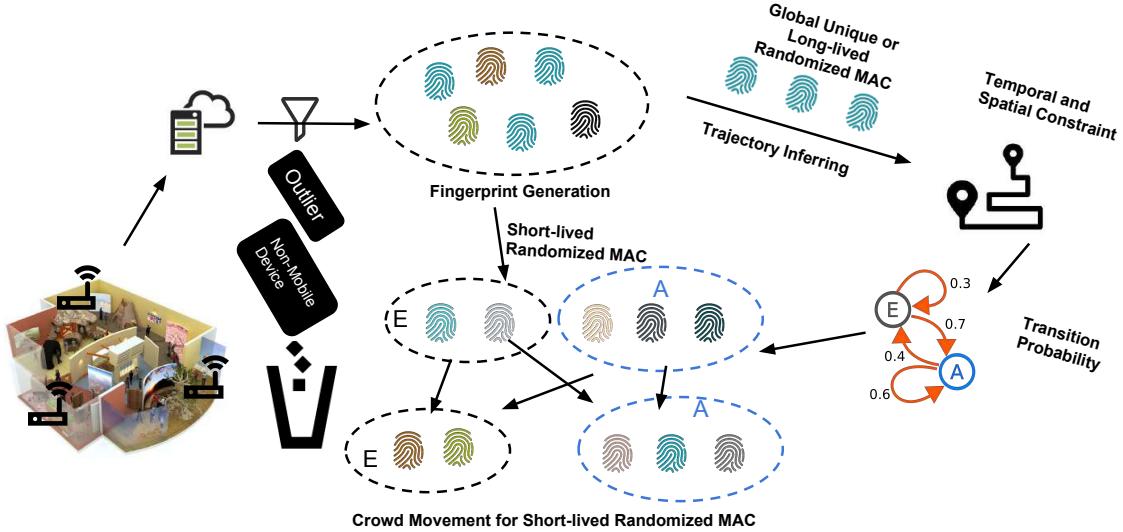


Fig. 2. Architecture of CrowdProbe

then the ratio of devices that have implemented randomization is close to 63% of the population. On average, each global unique MAC sent 34 probe request frames, while locally assigned MAC addresses were only sent with 5 probe request frames each. While global unique addresses have a 1-1 relationship with an individual device, a device performing randomization can have either 1-to-1 or 1-to-many relationship in a single day. Thus most of the randomized MAC addresses only existed in a limited number of probe request over a specific time period and were never seen again. Overall, we can see that randomized devices play an important role in crowd monitoring. If we are not able to properly tackle this problem, half of the information is concealed.

Table 1. Data statistics in Museum

Category	Global Unique MAC	Randomized MAC	Not mobile device
Probe Request Frame Number	1,744,764	3,006,941	108,262
MAC Address Number	50,953	602,133	2373
Probe request Per MAC	34	5	45

3 ARCHITECTURE AND DEPLOYMENT OF CROWDPROBE

The architecture of CrowdProbe is shown in Figure 2. Multiple WiFi monitors are deployed in varies locations and each WiFi monitor scans for probe request frames. When a user, carrying WiFi-enabled mobile devices, walks around different exhibition locations, the frames transmitted are captured by the monitors. Ideally, the WiFi monitors should be placed in the location that beacons from a device in any location within the monitored area can be heard by multiple monitors.

Data collected by the monitors are sent to the server for further processing. The server performs data analysis to generate crowd movement information:

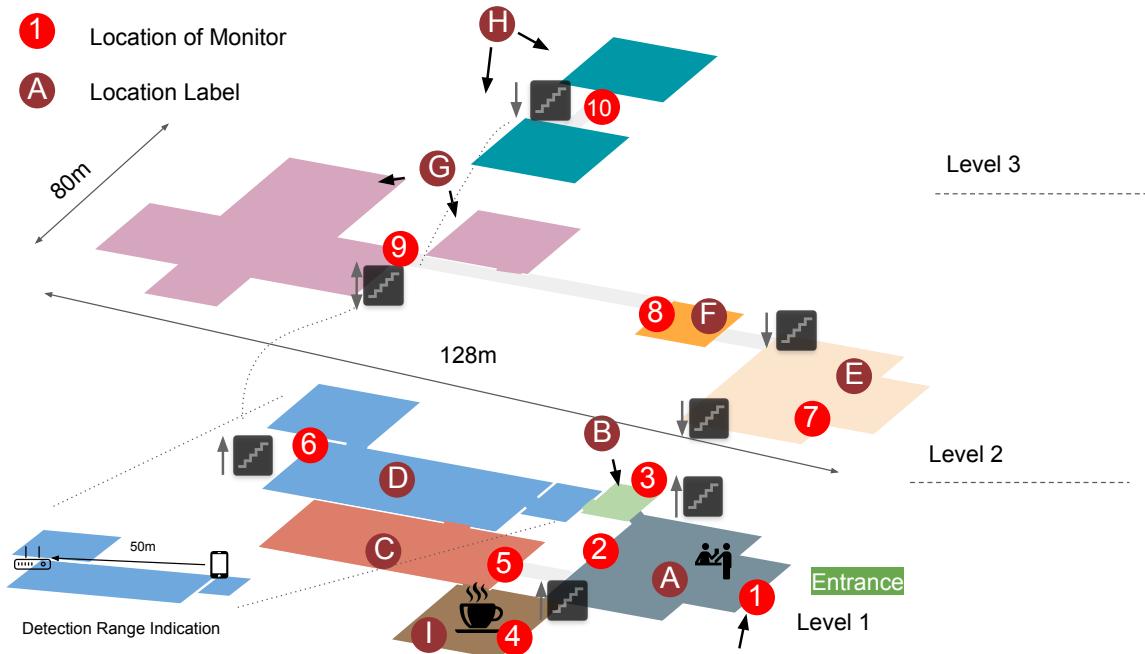


Fig. 3. Floorplan for the museum and the deployment layout

- **Device Filtering:** fingerprints from remote devices, non-mobile devices, and devices from staff in the museum are filtered out to make sure that the devices are carried by real visitors.
- **Fingerprint Generation and Classification:** probe request data from multiple monitors are merged to form the signal fingerprint. After that, these fingerprints are divided into two categories: stable MAC and short-lived randomized MAC.
- **Trajectory Inference:** Data from the stable set are used to generate visitors' trajectories based on temporal and spatial constraints. Using the result in trajectory generation, we are able to derive transition probabilities.
- **Movement Inference for Randomized Devices:** The transition probabilities can be input as a tool to guess the movement of randomized devices in a short time slot. By combining data from the randomized devices and global unique MAC devices, we can give a complete view of visitor movement in the museum.

The deployment of CrowdProbe has two components: the front-end WiFi monitors and back-end servers. We deployed the system in a museum of three floors. We divide the museum into 9 locations, marked with different colors in Figure 3. Location A is the main entrance and ticket counter. The location I is a cafe providing food and space for visitors to have a rest. The other seven locations are different exhibitions focus on different topics. The WiFi monitor deployed is a Raspberry Pi 3 device equipped with one D-Link wireless USB adapter(DWA-132). Raspberry Pi 3 is a low-cost computing platform with a 1.2 GHz quad-core ARM Cortex A53, 1 GB LPDDR2-900 SDRAM, and supports 802.11n Wireless LAN. Since the embedded WiFi adapter in the Raspberry Pi 3 cannot operate in the monitor mode, we instead use USB WiFi dongles to implement passive scanning. Each monitor can pick up transmissions sent by the mobile devices in the vicinity. Note that as the mobile devices transmit probes on all channels in the supported spectrum (typically both 2.4GHz and 5GHz), a monitor can ideally hear



Fig. 4. Monitors deployed in the museum

transmissions from all nearby mobile devices by sniffing on a single channel. However, in practice, due to packet loss, not all transmissions will be received. However, it has been noted that hopping between channel does not help to pick up more messages [13]. In our deployment, in order to maximize the probe request, the monitors are set to listen to the same channel with the nearby WiFi APs provided by the museum. To increase frame reception, we also sniff NULL data frames which are used for power management purpose when the devices are associated[15].

Figure 4 shows one of the monitors deployed and the device components we used for monitoring. We deploy a total of 10 boxes to ensure that we cover most of the exhibition locations. The deployment locations are labeled in Figure 3 with red circle icons. Due to aesthetic requirements by the museum management and the need to access power, we are not able to deploy the monitors in the desired locations to maximize coverage. Most of the monitors are installed under chairs, in corridors, or behind doors, which is not optimal for data collection. For example, in Figure 3, location C, and D do not have proper monitors in the center area. Nevertheless, we are able to cover most of the area sufficiently to understand visitors' movement pattern. The data collection is carried out with approval from the Institutional Review Board(IRB). To keep the privacy of visitors, we do not store the actual value but instead stored a hashed value of the MAC address after we verify that the MAC address is valid or randomized.

In the following sections, we will elaborate the details of each component of CrowdProbe and the corresponding challenges.

4 DEVICE FILTER AND CLASSIFICATION

In this section, we will describe the process carried out to increase the likelihood that the fingerprints collected come from visitors to the museum.

4.1 Filtering Remote Devices

Since the museum is located near a street famous for food and bars, monitors deployed may opportunistically capture probe frames from pedestrians on the streets. Such data has to be removed. This is handled by enforcing a minimum requirement of good quality RSS. While it is possible for visitors to visit signal blind spots with weak RSS, but it is not likely for visitors to spend all their time in such area. If the visitor walks around the museum, there is a good chance that strong RSS signals from the devices can be captured.

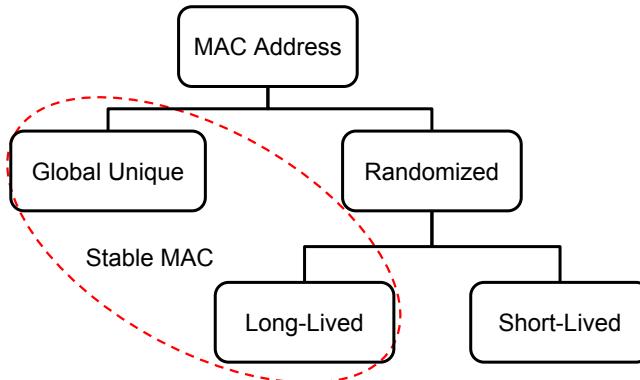


Fig. 5. MAC address classification

4.2 Filtering Non-mobile Devices

This filtering step is to make sure that the devices detected are from valid mobile device vendors. Note that we are mainly interested in smartphones carried by mobile users. We make use of the online public database to match the OUI of MAC addresses collected from probe request frames. Since we need to use the OUI field of the device, this step targets on global unique MAC address. Fortunately, non-mobile devices do not have the security concern to be tracked and thus lack the incentive to implement randomization.

4.3 Filtering Security Guard and Staff in Museum

Individuals inside the museum can be visitors or employees of the museum. The difference between these two categories is that visitors usually go to the museum occasionally, while employees stay in the museum over multiple days a week. Thus we keep a list of hashed MAC addresses that were captured by the monitors over multiple days. This set of devices may belong to employees in the museum or to non-mobile devices, for example, desktop with a WiFi dongle that comes from mobile phone vendor.

4.4 Fingerprint Generation and Classification

After device filtering, data collected from different monitors are merged into a signal fingerprint based on the time stamp. For example, the fingerprint is represented as $\vec{f}:\{r_1, r_2, r_3, \dots, r_n\}$, where r_i is the RSS captured by monitor i . We use the value of -99 to denote missing data when the monitor fails to capture the probe request or the monitor is too far away. Since all the monitors are connected to the internet to transmit data to the server, we also have to ensure that the clocks in monitors are well synchronized.

The last step to prepare the fingerprint data is as follows. We divide all the fingerprint data into two categories: stable MAC and short-lived MAC as shown in Figure 5. Stable MAC includes the global unique data and the set of randomized MAC data that do not change their mac address (Long-Lived). Long-Lived MAC addresses were sent by randomized devices, but they preserve the same randomized MAC over the entire visit. For these devices, we can track them as easily as the devices with globally unique MACs. Data from the globally unique and long-lived randomized MAC are given as input to generate the trajectories of visitors.

5 TRAJECTORY INFERENCE WITH HIDDEN MARKOV MODELS

To infer user movement trajectories, we model the visiting process as a probability-based state transition process. We adopt the most prevalent method used in passive tracking or indoor localization: Hidden Markov

Model(HMM)[10, 17]. HMM models next state based on the previous state, current observation, and transition probability. In our scenario, the hidden states are the location labels of visitors with given observations as RSS fingerprint vectors. Thus, we first try to match each fingerprint to a set of locations. Then we make use of spatial and temporal constraints to generate the transition probability and finally the target trajectory.

5.1 Emission Probabilities

The emission probability model defines the probability distribution of the visitors' location across the entire space where each fingerprint is captured. Correctly modeling the emission probability forms the basis for our trajectory inference. To make full use of the signal information in the passive RSS fingerprint from all the nearby monitors, we use fingerprint similarity to identify the location. We used four different phones to collect a fingerprint database in all the exhibition locations. We normalized the fingerprint and calculated their Tanimoto Coefficient[30]. Cross-validation is used to understand the performance of such fingerprint similarity method. The fingerprint database is separated into a training set and a testing set. We show the result of testing data from different phone models in Table 2. The four phone models (Nexus5, Nexus6, Meizu MX6, Meizu Pro6) are all using the Android OS. We use Android as we can easily modify the phone to send more probe frames with global unique MAC address.

Table 2. Classification result with different phone models

Train/Test	Mx6	Pro6	Nexus5	Nexus6
Mx6	0.88	0.68	0.66	0.72
Pro6	0.65	0.91	0.8	0.7
Nexus5	0.71	0.82	0.87	0.78
Nexus6	0.67	0.72	0.79	0.86

As we can see in Table 2, if the training set and testing set come from the same phone models, we are able to achieve close to 90% accuracy. However, when mapping to different phone models, the accuracy drastically drops to 70 percent. So, besides the multi-path effect, antenna gain and phone placement, phone model differences also have a negative impact on fingerprint matching. Furthermore, a phone can also transmit at different power levels depending on the specific IEEE 802.11 version used [11]. For example, Samsung Galaxy S4 sends at 13 dB using 802.11a but it sends at 12 dB using 802.11n.

Clearly, fingerprint similarity alone is insufficient to improve the accuracy. Our approach is to keep a set of locations in our emission probabilities. That is to say, we do not decide on a single location for each fingerprint. Instead, we keep a list of candidates assigning each of the possible candidate a probability. The idea is similar to particle filtering[9], but instead of keeping a large number of random sample, we only keep a limited set of candidates for higher efficiency. So for each fingerprint $f_i, \{r_1, r_2, \dots, r_n\}$, we have a list of candidate locations $\{l_1, l_2, \dots, l_n\}$. We calculate the similarity of each candidate with the corresponding fingerprint in the database. Then we get a list of fingerprint similarity $\{s_1, s_2, \dots, s_n\}$. We estimate the conditional probability of a device in location l_j given fingerprint f_i as follow:

$$\omega_j = (r_j + 99) / \sum_1^n (r_k + 99) \quad (1)$$

$$p(l_j | f_i) = \omega_j * s_j / \sum_1^n (\omega_k * s_k) \quad (2)$$

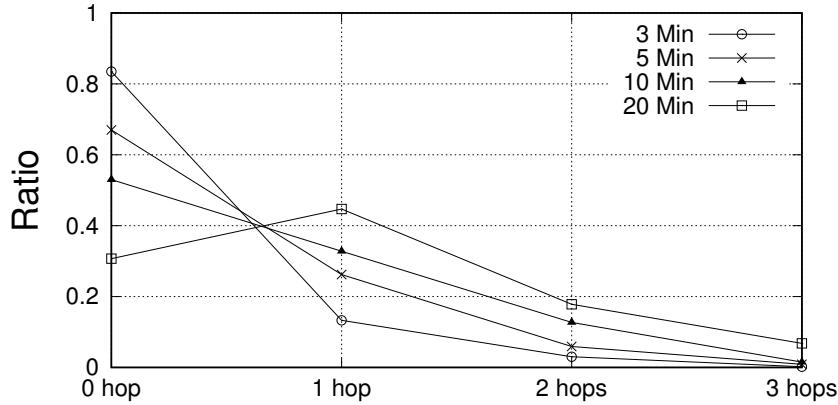


Fig. 6. Visitors' probabilities of moving to other locations with different time intervals

With the weight ω_j , we are giving more confidence to the stronger RSS fingerprints. After this step, for each fingerprint, we are able to generate a list of candidate locations and their emission probabilities. For example, {A: 0.7, B: 0.15, C: 0.07, E: 0.08} or {F: 0.26, G: 0.54, H: 0.21}.

5.2 Transition Probabilities

How the visitor will move between each pair of consecutive fingerprints is modeled as transition probabilities. We need to decide what is the probability for the visitor to move between exhibition locations or stay in the same location based on consecutive fingerprints and their time stamps. In our modeling, we made the following assumptions:

Assumption 1: A visitor's movement in the museum is sufficiently slow compared to the timescale of probe request capture such that the WiFi monitor is able to track his/her movement from one location to another.

For CrowdProbe to work well, a user needs to spend enough time in a single location so that it is likely that the device transmits at least one probe request from each location. To verify our assumption, we collected the transition pattern of visitors to the museum with different time intervals ranging from 3 min to 20 min and plotted the result in Figure 6. The x-axis shows how far a visitor can move, measured in the number of hops from current location to a destination location. From the figure, we can see that when the time interval is short, say 3 min, the likelihood that a visitor will stay in the same location is more than 80%. When the time interval is 20 min, a visitor has a 30% chance of moving to a location two or more hops away. In a 5 min interval, the likelihood of a visitor either staying in the same location or move to a neighboring location is 93%.

Assumption 2: The longer a visitor spends in an exhibition location, the more likely he will leave for the next exhibition location.

If a visitor has already spent some time, for example, 15 minutes, in the same exhibition location, then he is more likely to leave the location than the visitor who just arrives in this area. Thus, the transition probability should also take time already spent in the current location into consideration. Figure 7 shows the decay curve for locations D and E. As more time elapsed, more visitors will leave the place.

With this rule, we also solve a classical problem in passive tracking: the handover problem. The handover problem comes when the visitor is near the boundary of two different locations. The location inferred from the fingerprint can jump back-and-forth between the two locations. In our scenario, the museum is a multi-floor building where some of the ceilings between different floor are removed for aesthetic requirements. For instance, location A and E are connected openly without blocking, which leads to the problem that visitors in location A

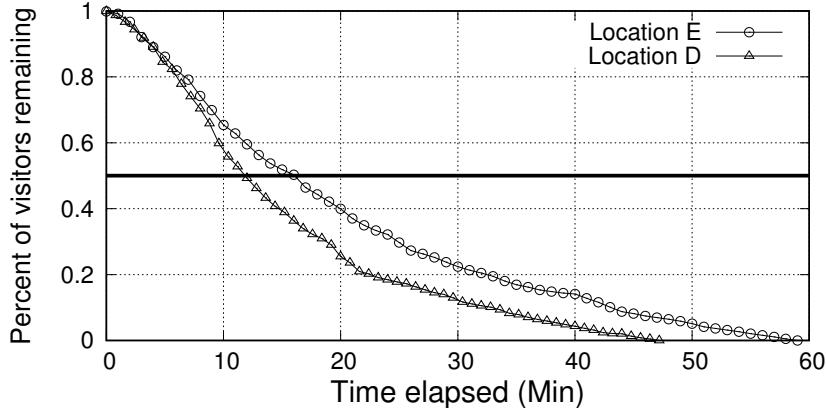


Fig. 7. The visitor decay curves for location D and E

have a high chance to be detected in location E. Sequences of fingerprints can generate jitters like AEAEAE in the trajectory derived. By taking the tendency to stay into consideration, more stable transitions can be obtained.

Based on the above discussion, we define the following:

Staying Tendency describes the inclination of the visitor to stay in the same exhibition location. From Figure 7, we can see that the percent of visitors remaining and the stay time follow an inverse proportional relationship. Thus we define the staying tendency coefficient ω_{tend} as

$$\omega_{tend} = \frac{\tau_{threshold}}{t + 1} \quad (3)$$

where t is the time length that the visitor has stayed in the current exhibition location. We add 1 to the ratio to handle extremely short duration t . The longer time the visitor has spent in the same location, the smaller the value of ω_{tend} will be. In Figure 7, we see different curves for different locations. Thus the time length threshold of $\tau_{threshold}$, which indicates the stay time length when a visitor has an equal chance to stay and leave the current location, should change based on different locations.

Order of Neighbor is defined as the number of locations a person must pass through to reach a specific exhibition location from the current location. For example, in the floorplan of the museum shown in Figure 3, for location C, the 1st-order neighbors include its immediate adjacent locations (A, D, and I), and its 2nd-order neighbors include the immediate adjacent locations of its 1st-order neighbors (excluding C and C's first order neighbors). We define hop_{ij} as the number of hops a person need to transit from location i to location j , which equal to the order of neighbor. Particularly, we set hop_{ii} as 1.

Based on the map constraint and temporal limitations, with time interval $\tau_{interval}$ between consecutive fingerprint, we define the transition likelihood $LH_{i \rightarrow j}$ and normalized transition probability $p_{i \rightarrow j}$ between location i and j as follows:

$$LH_{i \rightarrow j} = \begin{cases} \omega_{tend}/hop_{ii} + \tau_{interval}/\tau_{threshold}, & i = j \\ 1/hop_{ij} + \tau_{interval}/\tau_{threshold}, & otherwise \end{cases} \quad (4)$$

$$p_{i \rightarrow j} = \frac{LH_{i \rightarrow j}}{\sum_{k=1}^N LH_{i \rightarrow k}} \quad (5)$$

where N is the set of all the locations. With the increasing time interval between consecutive fingerprint $\tau_{interval}$, the relative difference of likelihood between each pair of locations becomes smaller. That means if the time interval between two fingerprints is small, we give higher transition probability to a nearby location. If the time interval is large we do not give any preference for the transition as the visitor can walk to any location within such a long duration. Table 3 gives the list of important parameter used.

Table 3. List of some important parameters in this paper

Parameter	Description
τ_{min}	The minimum staying time length required for visitor each location
s_i	RSS fingerprint similarity
ω_{tend}	staying tendency coefficient
$\tau_{threshold}$	Stay time length when visitor have a equal chance to stay and leave
$\tau_{interval}$	Time interval between consecutive fingerprint
hop_{ij}	The number of hops a person need to transit from location i to location j
$LH_{i \rightarrow j}$	Transition likelihood between location i and j
$p_{i \rightarrow j}$	Normalized transition probability between location i and j

5.3 Trajectory Inference

With the available transition probability and emission probability, we use Viterbi's algorithm[12] to find the maximum probability trajectory. For a series fingerprint f_1, f_2, \dots, f_n , we find the sequence locations l_1, l_2, \dots, l_n which maximize the Equation 6. Since we have only a limited number of candidates for each fingerprint captured, the result converges very fast. Usually, a visitor will spend quite a lot of time in a single location, thus the sequence of locations will contain a lot of redundancy. For example, AEEEEEEEFFFFFGGGGGGGGGGFADDI. Each letter represents the location of the visitor when a specific probe message was captured by the monitors. We simplify the trajectory by removing consecutive and duplicate locations and updating the corresponding time stamps. For the above example, we get AEFGFADI.

$$\operatorname{argmax}_{l_1, l_2, \dots, l_n} \prod_{i < n} p(l_{i+1} | f_i) * p_{(i \rightarrow i+1)} \quad (6)$$

6 ONE-HOP MOVEMENT INFERENCE FOR SHORT-LIVED RANDOMIZED DEVICE

From Table 1, we observe that if devices have similar probe frequencies, then the number of devices that implement MAC randomization is close to 2/3 of the population. While we are able to derive the crowd movement based on the above trajectory inference model using data from devices with Stable MAC addresses, ignoring a large number of devices with randomized MAC will lose a substantial amount of information. Previous work [31] used Information Elements (IE) as signatures to track devices. However, recent work[21] have found that such signatures may change during randomization. Improper use of IE may also cause a high rate of false positives. So if we can not track the movement of each randomized device, can we infer the crowd movement at each time duration without knowing who they are? In this section, we will use the trajectories derived from stable MAC devices to infer the one-hop crowd movement of short-lived randomized devices.

6.1 Overview

Figure 8 gives the overview of our one-hop movement inference for short-lived randomized devices. We cut the time duration in each day into separate time slots and generate the corresponding status vector in each

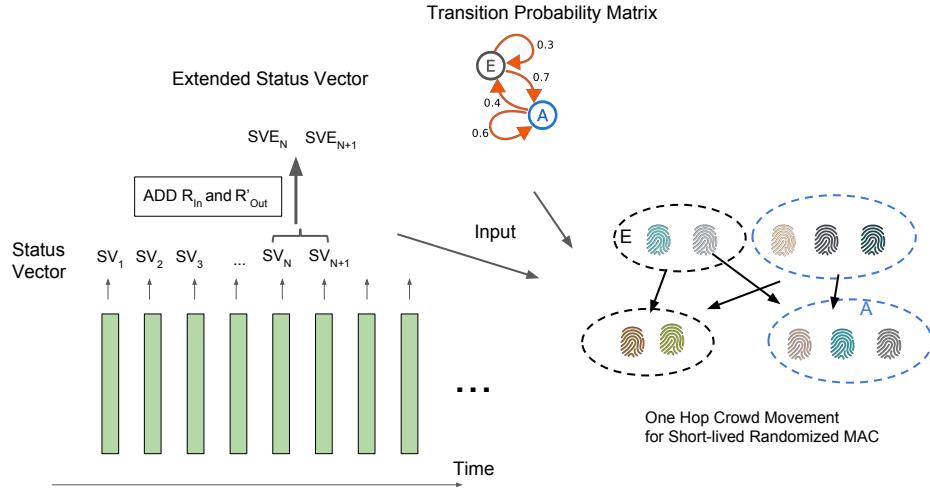


Fig. 8. Overview of the movement inference for short-lived randomized devices

time slot. A status vector is a vector that contains the number of randomized MAC devices which send probe frames captured in each location. A status vector is a snapshot of the number of short-lived MAC devices in each location. To complete the picture, we include the number of visitors who enter and leave the museum to form the extended status vector. R_{In} denotes the number of people entering the museum and R_{Out} denotes the number of people leaving the museum.

Although each individual visitor has his preference for route selection when visiting the museum, the choices are generally affected by the layout of exhibits, facilities, interpretative tools, and advertisements. If we assume that people carrying non-randomized phones have similar behavior to those carrying randomized phones, the aggregated movement should be similar. We utilize this assumption to infer crowd movement of users carrying devices with randomized MAC addresses based on the transition pattern learned from devices with Stable MAC addresses. In the next few sections, we will discuss the details of our algorithm.

6.2 Status Vector and Transition Matrix

Within each time slot, we define a status vector SV which contains the number of randomized probe frames captured in each location. An example of this will be $\{A : 10, B : 3, C : 3, D : 3, E : 1, F : 2, G : 3, H : 0, I : 2\}$. However, we found that this vector does not capture all the information about visitors that enter or leave the museum at the time. Since the museum has multiple entrances/exits and the probe transmission is opportunistic, a new visitor can appear or leave with last probe frame being captured in any location. Thus we define the extended version of status vector SVE to add two virtual locations, "In" and "Out". For every two consecutive vectors, we define the two $SVEs$ as follow:

where R_A and R'_A is the number of devices within location A in time slot N and N+1. We define the visitor movement between two time slots as a transition matrix T_N as follow:

Table 4. Extended Status Vector

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>In</i>	<i>Out</i>
<i>SVE</i> _{<i>N</i>}	<i>R</i> _{<i>A</i>}	<i>R</i> _{<i>B</i>}	<i>R</i> _{<i>C</i>}	<i>R</i> _{<i>D</i>}	<i>R</i> _{<i>E</i>}	<i>R</i> _{<i>F</i>}	<i>R</i> _{<i>G</i>}	<i>R</i> _{<i>H</i>}	<i>R</i> _{<i>I</i>}	<i>R</i> _{<i>In</i>}	0
<i>SVE</i> _{<i>N+1</i>}	<i>R'</i> _{<i>A</i>}	<i>R'</i> _{<i>B</i>}	<i>R'</i> _{<i>C</i>}	<i>R'</i> _{<i>D</i>}	<i>R'</i> _{<i>E</i>}	<i>R'</i> _{<i>F</i>}	<i>R'</i> _{<i>G</i>}	<i>R'</i> _{<i>H</i>}	<i>R'</i> _{<i>I</i>}	0	<i>R'</i> _{<i>Out</i>}

$$\begin{bmatrix} X_{A \rightarrow A} & X_{A \rightarrow B} & X_{A \rightarrow C} & \dots & X_{A \rightarrow H} & X_{A \rightarrow I} & 0 & X_{A \rightarrow Out} \\ X_{B \rightarrow A} & X_{B \rightarrow B} & X_{B \rightarrow C} & \dots & X_{B \rightarrow H} & X_{B \rightarrow I} & 0 & X_{B \rightarrow Out} \\ X_{C \rightarrow A} & X_{C \rightarrow B} & X_{C \rightarrow C} & \dots & X_{C \rightarrow H} & X_{C \rightarrow I} & 0 & X_{C \rightarrow Out} \\ \dots & \dots \\ X_{H \rightarrow A} & X_{H \rightarrow B} & X_{H \rightarrow C} & \dots & X_{H \rightarrow H} & X_{H \rightarrow I} & 0 & X_{H \rightarrow Out} \\ X_{I \rightarrow A} & X_{I \rightarrow B} & X_{I \rightarrow C} & \dots & X_{I \rightarrow H} & X_{I \rightarrow I} & 0 & X_{I \rightarrow Out} \\ X_{In \rightarrow A} & X_{In \rightarrow B} & X_{In \rightarrow C} & \dots & X_{In \rightarrow H} & X_{In \rightarrow I} & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \end{bmatrix}$$

where $X_{C \rightarrow C}$ denotes the number of visitors that remains in location C , $X_{A \rightarrow C}$ denotes the number of people that move from location A to C . Note that no visitor goes from state Out to any location, and no visitor goes from any location to state In . Both of these values in the matrix are set to 0. To conserve the number of people, all the variables need to satisfy the following equations:

$$\left\{ \begin{array}{l} X_{A \rightarrow A} + X_{A \rightarrow B} + X_{A \rightarrow C} + \dots + X_{A \rightarrow Out} = R_A \\ \dots \\ X_{I \rightarrow A} + X_{I \rightarrow B} + X_{I \rightarrow C} + \dots + X_{I \rightarrow Out} = R_I \\ X_{In \rightarrow A} + X_{In \rightarrow B} + X_{In \rightarrow C} + \dots + X_{In \rightarrow Out} = R_{In} \\ X_{A \rightarrow A} + X_{B \rightarrow A} + X_{C \rightarrow A} + \dots + X_{In \rightarrow A} = R'_A \\ \dots \\ X_{A \rightarrow I} + X_{B \rightarrow I} + X_{C \rightarrow I} + \dots + X_{In \rightarrow I} = R'_I \\ X_{A \rightarrow Out} + X_{B \rightarrow Out} + X_{C \rightarrow Out} + \dots + X_{In \rightarrow Out} = R'_{out} \\ R_{In} - R'_{out} = R_{gap} \end{array} \right. \quad (7)$$

Now based on the processing of randomized MAC data, we can derive values R_A to R_I , R'_A to R'_I and R_{gap} . Suppose we have N locations in the museum (not including In and Out). We have $2 \times N + 3$ equations in the above formulation. However, we have $N \times N + 2 \times N$ unknown values. Whenever $N > 1$, we have $N(N + 2) > 2N + 3$. So there exist many different transition matrices that can satisfy the equations. Thus, we need a way to find a specific solution that satisfies additional constraints. Our approach is to make use of the data accumulated with global unique and long-lived randomized address. The approach uses a two-step process to infer the one-hop movement for the short-lived randomized device.

6.3 Two-steps Conversion for Short-Lived Randomized Data

We assume that people carrying non-randomized phones have similar moving patterns with those carrying randomized phones. In order to utilize such movement pattern, we perform the same processing to the stable MAC data set as described in section 6.2. With every two consecutive time slots, we are able to get one ground-truth transition matrix since these devices keep the same MAC addresses. We sum up all the transition matrices and normalize each row to generate the probability matrix T_{train} . We assume that T_{train} captures the average user behavior.

Thus in the first step, SVE_N is multiplied by the transition matrix T_{train} to generate the expected status vector SVE'_N .

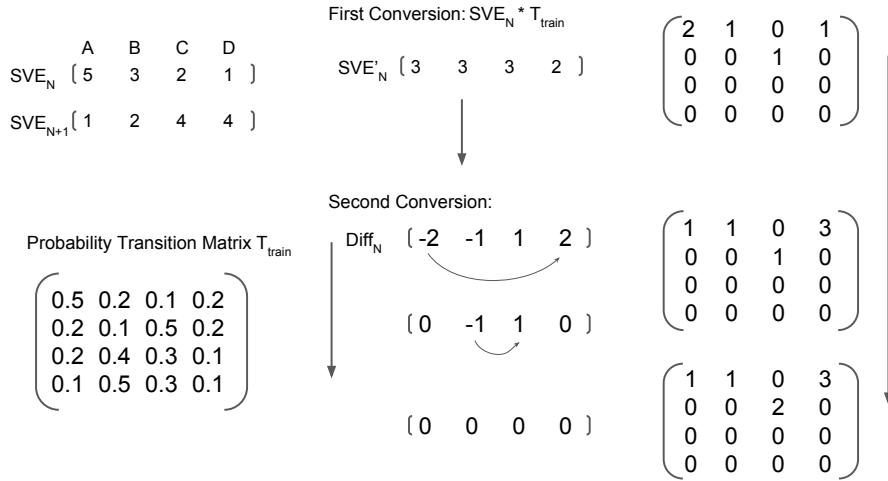


Fig. 9. A simple example for two-step conversion

$$SVE'_N = SVE_N * T_{train} \quad (8)$$

As the current occupancy status can be different from the average behavior, in the second step, we find the status vector that is close to SVE'_N and yet minimizes the differences. For ease of processing, we calculate the difference vector $Diff_N$ by subtracting SVE'_N from SVE_{N+1} .

$$Diff_N = SVE_{N+1} - SVE'_N \quad (9)$$

Each of the negative values in $Diff_N$ indicates that there is a certain number of visitors that move from current location to the other locations. For each positive value of $Diff_N$, it means this location attracts visitors from other locations. Thus, with each negative value, we search for available positive values in the $Diff_N$ to fill the hole. Based on the transition probability, we assign the visitors to move to the other corresponding location until the $Diff_N$ is adjusted to be a vector containing all 0 values. With these sequence of conversions, we finish the second step conversion.

The final transition matrix gives an estimation of the crowd movement for people bringing devices with randomized MAC during this period. Figure 9 gives a simple example of the two-step conversion calculation with only four locations included without In and Out state for ease of explanation. Summing up the matrix for both Stable MAC and Short-lived Randomized MAC data set, we have an overview of the crowd movement for visitors.

7 EVALUATION

There are three parts in the evaluation. First, we evaluate the accuracy of trajectory inferring using Stable MAC data. We then present the results for inferring movement of devices with random MAC addresses using movement patterns of devices with Stable MAC addresses. Finally, we present some interesting crowd movement statistics findings in the museum.

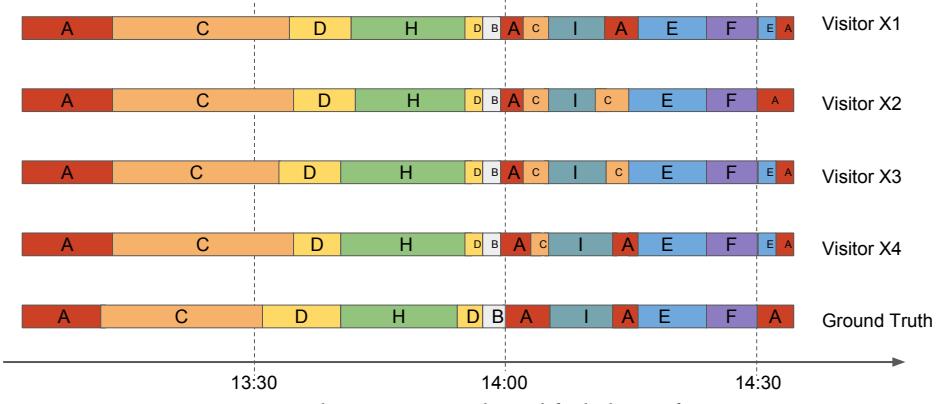


Fig. 10. Path generation with modified phones for TR1

7.1 Accuracy of Trajectory Inferring

Two parameters are required in the inference. First, we need the minimum staying time length τ_{min} that a visitor has to spend in a location for the system to be able to detect the movement. This duration depends on how frequent each device transmits the probe frames. Based on measurements presented earlier, the parameter is set to 5 min.

The second parameter needed, $\tau_{threshold}$, indicates the stay time length when a visitor has an equal chance to stay and leave. We use the average staying duration in each of the locations to infer the value. Since different locations have different area sizes, $\tau_{threshold}$ can vary a lot for different areas. The corresponding value of $\tau_{threshold}$ measured for different locations are {A : 18min, B : 9min, C : 16min, D : 13min, E : 13min, F : 10min, G : 32min, H : 13min, I : 13min}.

7.1.1 Ground Truth Collection. To verify the accuracy of trajectory inference, we organize three ground truth collection tours to the museum. The details of the tours are listed in Table 5. In TR1, we organized four people carrying four different phones to walk on a predefined route. All the phones in this tour are modified to prompt more probe request frames with global unique MAC address. We required the users to record down the time stamps in each of the locations. In TR2, we followed a one-hour guided tour with 11 other adult visitors (16 in total including tour guide). All the visitors just use their mobile devices which are unmodified phones with WiFi switched ON. In TR3, we do a similar guide tour with more young people involved.

Table 5. Detail of three ground truth collection tour

Name	Number of People	Young	Old	Identified	Visiting Route	Time
TR1	4	4	0	4	ACDHDBIAEFA	1h 30min
TR2	16	7	9	9	ACDH	55min
TR3	18	13	5	13	ACDBAEF	1h 8min

7.1.2 Trajectory Inferring Result with Probe Frequency. We will only show the result of the first two trips as the result of the third trip is similar. The result of the path inference for TR1 is shown in Figure 10 with ground truth plotted in the bottom. We found that the trajectories inferred are pretty accurate with minor errors. The

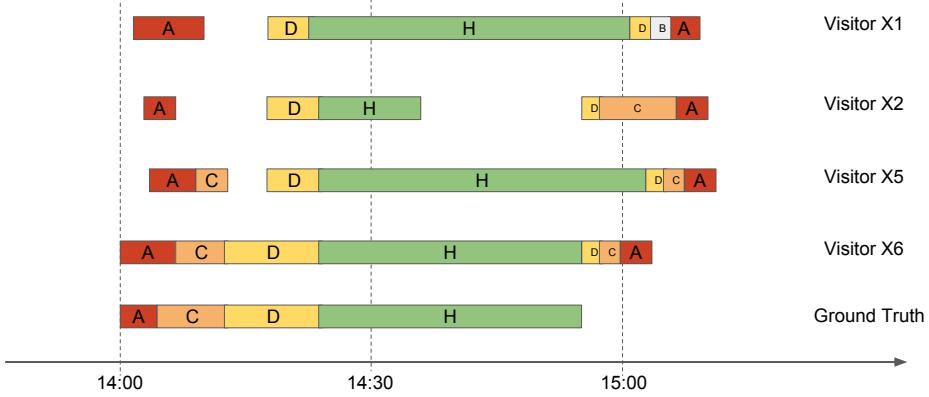


Fig. 11. Path generation with unmodified phones for TR2

start time and end time for each location differ from the ground truth by a maximum of 3 minutes. The reason for such high accuracy is because of the use of the modified phones with a high frequency of probe transmission.

The result in Figure 11 shows the ground truth and our trajectory inference for four of the visitors during TR2. The ground truth tour ended after visiting location H. We include the full traces of the four visitors which showed their personal choices after the tour guide ended the tour. Compared to the result in Figure 10, only X6 generate the full trajectory without any gap. Trajectories for visitor X1 and X2 both miss the location C, which may be because they stayed there only for a short time. In Visitor X2's trajectory, there is a duration of around 20 minutes without any probe request emitted for which we can not decide on the proper location. While we can guess that the visitor may remain in the same location H, such an assumption may lead to a large error. Thus we leave that period of time as unknown.

7.1.3 Trajectory Inferring Accuracy. We define several metrics to measure the accuracy of trajectory inference:

- **False Positive** The ratio of the locations identified by the algorithm in the trajectories but not present in ground truth trajectories.
- **Location Recall** The correct number of locations derived in trajectory / The total number of locations in ground truth.
- **Time Length Accuracy** Time length estimation accuracy.
- **Start and End Time Error** The start time and end time shift errors for each location we identified in the trajectory.

Based on the result of the three trips, we identified 26 trajectories and use them to calculate the accuracy of trajectories inference. We compare the performance of three approaches used to derive trajectory.

- **FP** uses only the WiFi fingerprinting method for localization and uses these locations to derive the trajectories.
- **HMM** is similar to CrowdProbe but without considering the movement pattern of visitors inside the museum. That is to set the transition probability as same for all the locations.
- **CrowdProbe** the proposed method.

The results are shown in Figure 12 and Figure 13. It can be observed that CrowdProbe attains a low false positive close to 0.14 compared to 0.23 for FP and 0.18 for HMM. While CrowdProbe has similar recall rate with HMM

method, the time length estimation for CrowdProbe is much higher at 0.94. The Start Time and End Time Error for FP and HMM are around 5 minutes and 3 minutes. CrowdProbe improves that to around 2.5 minutes. The improvement is due to reducing the jitters in the handover area.

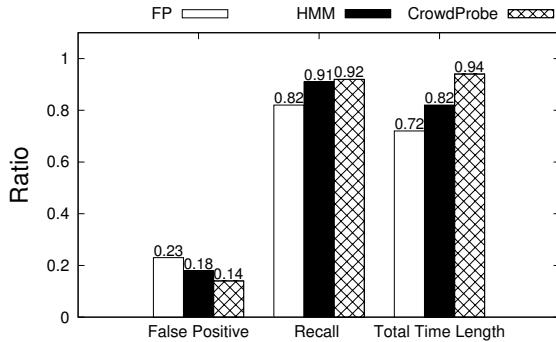


Fig. 12. Trajectory generation performance

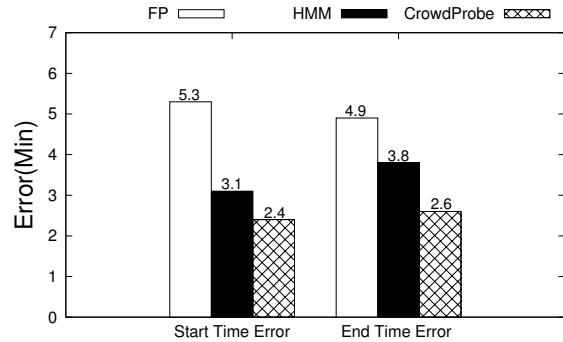


Fig. 13. Time stamp estimation performance

7.2 Short-lived Randomized Device One-hop Transition Evaluation

7.2.1 Time Slicing. We need to decide what is the proper duration or time slot length that we are able to infer without losing too much information about crowd movement but with sufficient collected data. The basic requirement for picking the length of the time slot is to ensure that device with randomized MAC will send at least 1 probe request in each time slot but not multiple probe frames with different MAC addresses. Thus this value is decided by two factors: the lifetime of randomized MAC and probe frequency.

The only hint we found about the lifetime of randomized MAC is in the configuration file *wpa_supplicant.conf* used by Android and Windows and OS client station which indicate a *rand_addr_lifetime* = 60 [20]. That means any two randomized addresses are not likely to emitted from the same device within 1 minute. Thus, we can safely set the time slot length to be larger than 1 minute. From Figure 1, we can see that most of the devices send at least one probe frame within a 5-minute time slot. With a larger value like 10 minutes, we are likely to include multiple samples from the same device in each time slot. That may introduce error in the status vector. With a much smaller value like 1 minute, we may not have received any probes from many of the devices.

A device may transmit more than one probe frame in the same 5-minute slot. If the (randomized) MAC address remains the same, then this is not a problem. However, if the MAC address changes within a single time slot, then the same device may be counted as different devices and we overestimate the number of users. Hence, a relatively short interval of 5 minute will also limit the amount of overestimation due to duplicates.

7.2.2 Evaluation Method. Even though we can derive the transition matrix for randomized devices, we are not able to verify the result since probe frames are randomized as we do not have the ground truth data for the short-lived randomized MAC data. Thus, we instead use the data from Stable MAC devices to check the performance of our approach. The flow of the evaluation is given in Figure 14. We input the status vectors SVE_N and SVE_{N+1} for the Stable MAC devices to the algorithm and get the result transition matrix. With the Stable MAC data set, we can easily derive the ground truth transition matrix. We use the following metrics to measure the performance of the short-lived MAC device one-hop transition inference.

- **Transition Accuracy** The number of correct transitions in Transition matrix for Stable MAC data / The total number of transition happen in ground truth. Note $A \rightarrow A$ is also regarded as one transition.

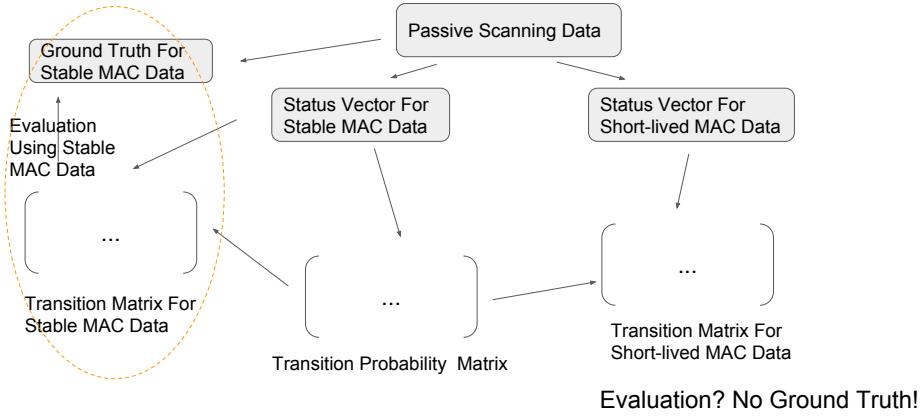


Fig. 14. Demonstration of our evaluation method for one-hop transition inference

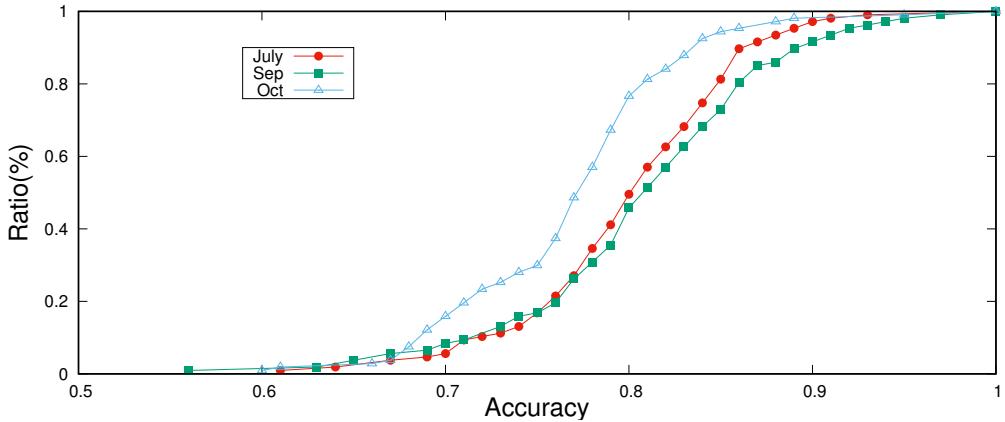


Fig. 15. Randomized trajectory inference accuracy with non-randomized data testing result

7.2.3 Result for Short-lived Randomized Device Transition. By comparing the ground truth matrix and the one we derived, we can estimate the performance of our Short-lived transition inference method. With each two consecutive time slots, we run the evaluation method to verify the effectiveness of the short-lived randomized device one-hop transition. We show the results for the month of July, September, and October 2017 by plotting the CDF of the transition accuracy in Figure 15. The average accuracy for the three months is 0.8, 0.81, 0.77 respectively. That means in every 5 transitions, we can correctly infer 4 of them. Considering the difficulty of tracking devices with randomized MAC addresses, the accuracy is better than what we expected.

Table 6 gives a summary of the information we can get from passive scanning. If the device provides Stable MAC addresses, we can derive a lot of information about the crowd movement. We can derive short time movement for devices with randomized MAC addresses if we can supplant the data with statistics of stable MAC in the same venue. However, if the information is only for short-lived randomized MAC, we can only do occupancy counting in each time slot.

Table 6. Information we can obtain from passive scanning

Feature	Counting	R_{In} and R_{Out}	One-hop Transition	Trajectory inference	Stay Length Estimation
Stable MAC	✓	✓	✓	✓	✓
Short-Lived MAC with Stable MAC statistics	✓	✓	✓	✗	✗
Short-Lived MAC Only	✓	✓	✗	✗	✗

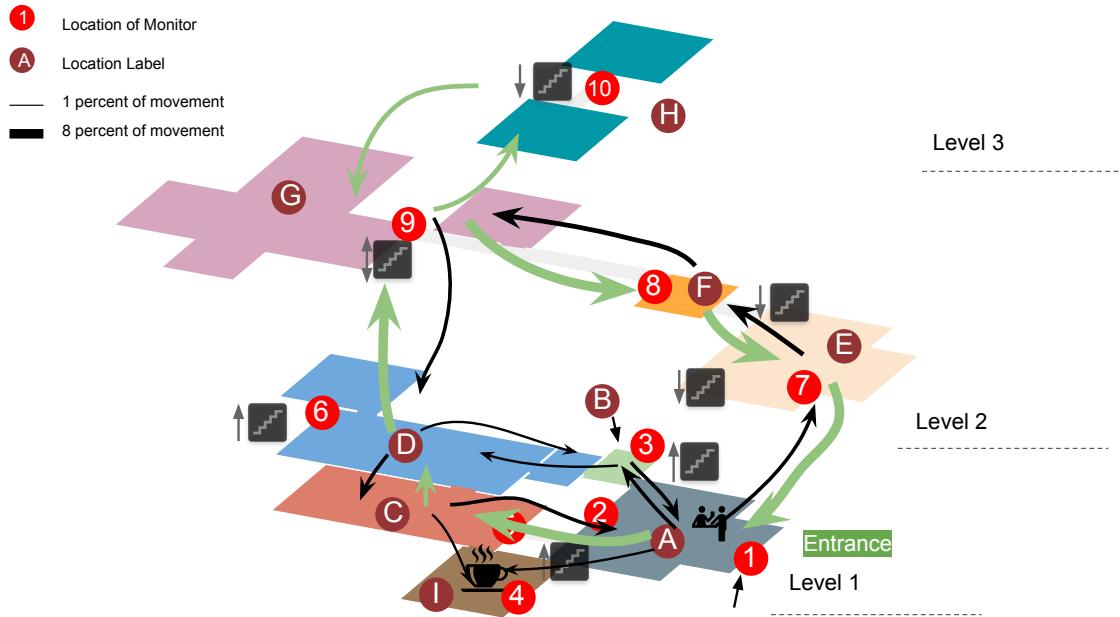


Fig. 16. The arrows and their widths represent visitors' flows between different locations. The most frequent path is shown as green color.

7.3 Findings for Museum Statistics

With the help of the trajectory and transition inference algorithm, we share our findings in processing museum data regarding the visitors' movement pattern. In August 2017, the layout of the museum was changed due to some artwork being replaced and exhibition location G being blocked for re-installation. Thus in our analysis, we may also include the impacts of such changes.

Although each visiting path selection can be affected by personal choice, from a macro view, the path spatial distribution should be the result of the interplay between monitors locations and the spatial layout of the museum. Figure 16 gives the spatial distribution of visitors in the museum. From the figure, we can see a majority amount of visitors follow the route ACDGHFEA and a smaller set of visitors take the reverse route with AEFGHDCA. The two paths both begin from and end at location A which is the main entrance to the museum. Among all the sub-path, EFG and GFE appear in 35% and 29% of all the visitors' trajectories. This is because of the linear layout on the second floor of the museum. The number of visitors picks ACD is twice the number of visitors who pick ABD. Only 18% of people actually make a visit to exhibition location B. 73% of the visitors actually

skip exhibition location H which located deep inside the museum in the third floor. CrowdProbe enables us to analysis on such visitor pattern without labor-intensive survey.

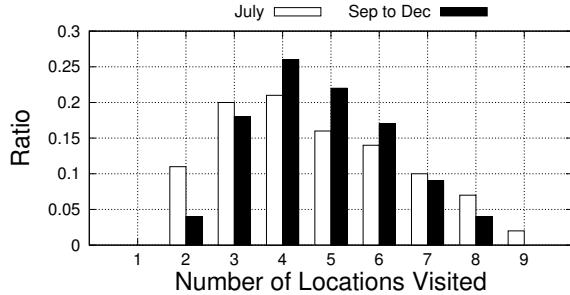


Fig. 17. Number of exhibition locations visited for visitors

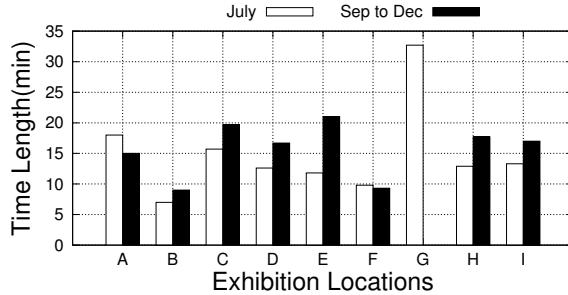


Fig. 18. Average staying time for each exhibition location

Figure 17 shows the average number of exhibition locations visited. 80% of the visitors took a tour including 3-6 locations. Only very few visitors actually visit the whole museum. After the re-innovation started in August, the average number of locations visited decreases slightly. Figure 18 gives the average staying time for each location. The duration usually ranges from 10-20 minutes and is somewhat proportional to the area of each location. With the change in August, the time spent in location G is distributed to other exhibition locations causing an increment of staying in almost all the other areas.

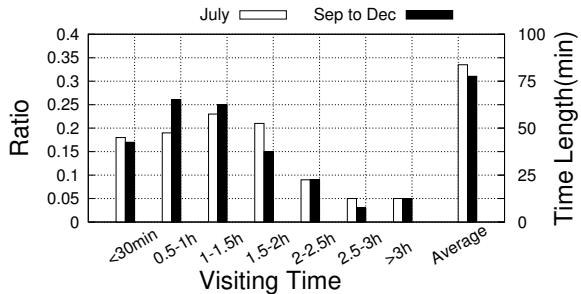


Fig. 19. Distribution of total time length spend in museum

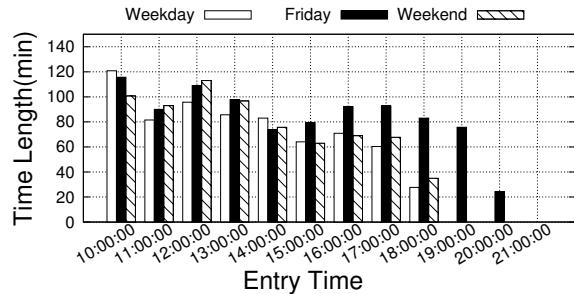


Fig. 20. Visitor stay time length vs entry time

Figure 19 shows the total time distribution a visitor spent in the museum. 63% and 66% of visitors spent around 0.5 to 2 hours at the museum in July and Sep-Dec. Only five percent of the visitors spent more than 3 hours in the museum. After access to location G had been blocked, the time in the museum drops slightly from 84min to 79min. From Figure 20, we can conclude that the visitor will stay for shorter length when reaching the closing time for the museum (21:00 on Friday, 19:00 otherwise). In the morning, around 11 am, visitors tend to stay less time, which may be because of the approaching lunchtime. In the afternoon, the duration is relatively stable and starts decreasing two hours before the closing time.

8 RELATED WORK

Research work on passive tracking can generally be divided into two categories: device-free passive tracking and device-based passive tracking.

8.1 Device-free Passive Tracking

The idea of radio-based device-free passive tracking is based on the fact that the existence of the human body in an RF environment affects the RF signals, especially in 2.4 GHz and 5 GHz band common in WiFi network. Typical deployment usually includes signal transmitters and monitoring points. During the training phase, RSS information is collected under different conditions. Later in the testing phase, the emerging RSS fingerprint is matched to the database to infer the number of people and their locations. While this technology is still only used in controlled experiment settings, some research work already shows the potential. Nuzzera[28] used probabilistic approach for handling the device-free passive localization problem for a single intruder. E-eye[34] uses channel state information to identify and distinguish in-house activities. Ichneaa[27] shorten the training period and applied statistical anomaly detection techniques and particle filtering to provide localization capabilities.

Device-free passive tracking usually requires tedious training and can only track a limited number of people. Moreover, if the environment settings have changed, the radio database needs to be re-trained and adjusted to fit the new changes. This limitation hinders the further deployment of such device-free passive tracking technology.

8.2 Device-based Passive Tracking

Device-based passive tracking aims to track devices that carried with users, especially smartphones. Early work[18] uses RFID to estimated visitor positions, visiting patterns, and inter-human relationships at a science museum. Recent work[35] use Bluetooth to monitor visitors' length of stay at the Louvre. However, their experiment only cover 8.2% of the visitor which affects the credibility and practicality of the conclusion. Due to the widespread deployment of WiFi networks and the popularity of smartphone, the use of WiFi related information to individual information has been both popular and shown to be effective. Researchers have proposed a series of ideas to exploit the availability of probe information from mobile devices to track individuals. Musa[23] also used HMM-based method to estimate smartphone trajectory which is similar to our work. However, their system is meant to deploy for outdoor road conditions where vehicle have fix moving direction and the requirement for granularity is lower than in a complex museum.

Besides merely tracking location, more work focuses on revealing user relationship such as using the known SSIDs list in probe requests as the fingerprint to decide whether two people are socially linked together[3, 7]. A similar method has been used to generate spatial-temporal similarity based on users' co-occurrence frequency to infer relationships between them [5, 15]. Adriano[8] exploited WiFi probe requests to de-anonymize the origin of participants in large events. To combat such information leakage, major mobile phone vendors introduced MAC randomization and encouraged the devices to send probe frames with empty (unknown) SSID list [16].

After the introduction of MAC randomization, researchers focus on de-anonymize WiFi frame. Freudiger[13] attempt to use sequence number and timing information to link randomized probe message. Vanhoef[31] make use of information element(IE) and scrambler seeds used at the physical layer to track users. Martin[21] is more aggressive to implement control frame attack to expose the globally unique MAC.

Compared to the previous works, CrowdProbe is deployed in a complex indoor environment. We provide a non-invasive method to reveal the crowd movement regardless of the phone vendors or OS versions.

9 DISCUSSION

In our measurements, we observe that about 60% of the devices randomized their MAC addresses. As more vendors take action to protect the privacy of the user, this ratio will continue to increase. While such a trend presents a challenge for CrowdProbe, we would like to highlight that CrowdProbe can work as long as there is sufficient statistics from devices that broadcast frames with Stable MAC addresses. We believe this will be the case for the following reasons. First, if a device is associated with the WiFi network, it will revert to use its global unique MAC address [21]. In many public spaces, free WiFi access is often available. It can be expected that some

visitors will connect to the WiFi network for internet access. Thus, one will be able to collect sufficient statistics with Global Unique MAC even though it may take more time. Second, as shown in Figure 5, some devices indeed randomized their MAC, but they keep the same randomized MAC over a sufficiently long duration of up to hours. Such data can be used to infer the transition probability without linking each MAC to a specific user device. Lastly, we also sniff NULL data frame. These frames are used for power management and do not randomized the MAC addresses. Current randomization scheme is implemented only on the active scanning of the mobile device. Based on the above discussion, CrowdProbe can continue to collect enough data to infer the transition pattern and one-hop transition for devices with randomized MACs.

While CrowdProbe is only deployed and tested in the museum environment, the technique has the potential to be used in other environments like shopping malls and transportation hubs. For trajectory inferring, all the parameters are based on the data collected in the place. Thus, our algorithm will still run for the different scenarios as long as sufficient data can be collected.

The sparse nature of frames transmission limits the accuracy of crowd monitoring which can be seen from the performance gap between Figure 10 and Figure 11. To get more frame transmission, the author in [23] propose to emulate the SSID of popular or previously visited AP. This technique can also be integrated into our system. However, this technique triggers the use of WiFi interface of the mobile device which will interrupt the existing connection and drain the battery at a higher rate.

10 CONCLUSION

In this paper, we propose an HMM-based visitor trajectory inference method based on passive WiFi monitoring. Moreover, we make use of the transition probability derived from existing trajectories to generate the possible movement mapping. The deployment and evaluation in a multi-floor museum proved the feasibility of the proposed system. We believe that CrowdProbe can also be used in other scenarios. While there is no fixed model for all the applications, the experience and lessons we learn from this case study will help in bridging research and practice.

REFERENCES

- [1] Lada A Adamic and Eytan Adar. 2003. Friends and neighbors on the web. *Social networks* 25, 3 (2003), 211–230.
- [2] Jamal Jokar Arsanjani, Wolfgang Kainz, and Ali Jafar Mousivand. 2011. Tracking dynamic land-use change using spatially explicit Markov Chain based on cellular automata: the case of Tehran. *International Journal of Image and Data Fusion* 2, 4 (2011), 329–345.
- [3] Marco V Barbera, Alessandro Epasto, Alessandro Mei, Vasile C Perta, and Julinda Stefa. 2013. Signals from the crowd: uncovering social relationships through smartphone probes. In *IMC*. ACM, 265–276.
- [4] Ben Benfold and Ian Reid. 2011. Stable multi-target tracking in real-time surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 3457–3464.
- [5] Ningning Cheng, Prasant Mohapatra, Mathieu Cunche, Mohamed Ali Kaafar, Roksana Boreli, and Srikanth Krishnamurthy. 2012. Inferring user relationship from hidden information in wlans. In *MILITARY COMMUNICATIONS CONFERENCE, 2012-MILCOM 2012*. IEEE, 1–6.
- [6] Tom Chothia and Vitaliy Smirnov. 2010. A Traceability Attack against e-Passports.. In *Financial Cryptography*, Vol. 6052. Springer, 20–34.
- [7] Mathieu Cunche, Mohamed Ali Kaafar, and Roksana Boreli. 2012. I know who you will meet this evening! linking wireless devices using wi-fi probe requests. In *WoWMoM*. IEEE, 1–9.
- [8] Adriano Di Luzio, Alessandro Mei, and Julinda Stefa. 2016. Mind your probes: De-anonymization of large crowds through smartphone WiFi probe requests. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*. IEEE, 1–9.
- [9] Arnaud Doucet, Nando De Freitas, Kevin Murphy, and Stuart Russell. 2000. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 176–183.
- [10] Sean R Eddy. 1996. Hidden markov models. *Current opinion in structural biology* 6, 3 (1996), 361–365.
- [11] Samsung Electronics. 2013. SAR evaluation report. In *SAR evaluation report*. Samsung Electronics, 3–4.
- [12] G David Forney. 1973. The viterbi algorithm. *Proc. IEEE* 61, 3 (1973), 268–278.

- [13] Julien Freudiger. 2015. How talkative is your mobile device?: an experimental study of Wi-Fi probe requests. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 8.
- [14] Dan Goodin. 2017. Shielding MAC addresses from stalkers is hard and Android fails miserably at it. <https://arstechnica.com/information-technology/2017/03/shielding-mac-addresses-from-stalkers-is-hard-android-is-failing-miserably/>. [Online].
- [15] Hande Hong, Chengwen Luo, and Mun Choon Chan. 2016. SocialProbe: Understanding Social Interaction Through Passive WiFi Monitoring. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ACM, 94–103.
- [16] Xueheng Hu, Lixing Song, Dirk Van Bruggen, and Aaron Striegel. 2015. Is There WiFi Yet? How Aggressive WiFi Probe Requests Deteriorate Energy and Throughput. *arXiv preprint arXiv:1502.01222* (2015).
- [17] Mohamed Ibrahim and Moustafa Youssef. 2011. A hidden markov model for localization using low-end GSM cell phones. In *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 1–5.
- [18] Takayuki Kanda, Masahiro Shiomi, Laurent Perrin, Tatsuya Nomura, Hiroshi Ishiguro, and Norihiro Hagita. 2007. Analysis of people trajectories with ubiquitous sensors in a science museum. In *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 4846–4853.
- [19] Thomas Liebig and Armel Ulrich Kemloh Wagoum. 2012. Modelling Microscopic Pedestrian Mobility using Bluetooth.. In *ICAART (2)*. 270–275.
- [20] Jouni Malinen. 2014. Linux WPA/WPA2/IEEE 802.1X Supplicant. https://w1.fi/wpa_supplicant/. [Online].
- [21] Jeremy Martin, Travis Mayberry, Collin Donahue, Lucas Foppe, Lamont Brown, Chadwick Riggins, Erik C Rye, and Dane Brown. 2017. A Study of MAC Address Randomization in Mobile Devices and When it Fails. *arXiv preprint arXiv:1703.02874* (2017).
- [22] Lyudmila Mihaylova, Paul Brasnett, Nishan Canagarajah, and David Bull. 2007. Object tracking by particle filtering techniques in video sequences. *Advances and challenges in multisensor data and information processing* 8 (2007), 260–268.
- [23] ABM Musa and Jakob Eriksson. 2012. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM conference on embedded network sensor systems*. ACM, 281–294.
- [24] Nuria M Oliver, Barbara Rosario, and Alex P Pentland. 2000. A Bayesian computer vision system for modeling human interactions. *IEEE transactions on pattern analysis and machine intelligence* 22, 8 (2000), 831–843.
- [25] Oluwatoyin P Popoola and Kejun Wang. 2012. Video-based abnormal human behavior recognition—A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 6 (2012), 865–878.
- [26] Romer Rosales and Stan Sclaroff. 1999. 3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on*, Vol. 2. IEEE, 117–123.
- [27] Ahmed Saeed, Ahmed E Kosba, and Moustafa Youssef. 2014. Ichnaea: A low-overhead robust WLAN device-free passive localization system. *IEEE Journal of selected topics in signal processing* 8, 1 (2014), 5–15.
- [28] Moustafa Seifeldin, Ahmed Saeed, Ahmed E Kosba, Amr El-Keyi, and Moustafa Youssef. 2013. Nuzzer: A large-scale device-free passive localization system for wireless environments. *IEEE Transactions on Mobile Computing* 12, 7 (2013), 1321–1334.
- [29] K. Skinner and J. Novak. 2015. Privacy and your app. [Online].
- [30] Taffee T Tanimoto. 1958. Elementary mathematical theory of classification and prediction. (1958).
- [31] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S Cardoso, and Frank Piessens. 2016. Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 413–424.
- [32] Mathias Versichele, Tijs Neutens, Matthias Delafontaine, and Nico Van de Weghe. 2012. The use of Bluetooth for analysing spatiotemporal dynamics of human movement at mass events: A case study of the Ghent Festivities. *Applied Geography* 32, 2 (2012), 208–220.
- [33] Harald Vogt. 2002. Efficient object identification with passive RFID tags. *Pervasive computing* (2002), 98–113.
- [34] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. 2014. E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures. In *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 617–628.
- [35] Yuji Yoshimura, Anne Krebs, and Carlo Ratti. 2017. Noninvasive Bluetooth Monitoring of Visitors' Length of Stay at the Louvre. *IEEE Pervasive Computing* 16, 2 (2017), 26–34.

Received February 2018; revised July 2018; accepted September 2018