Indoor Occupancy Tracking in Smart Buildings Using Passive Sniffing of Probe Requests

Edwin Vattapparamban*, Bekir Sait Çiftler*, İsmail Güvenç*, Kemal Akkaya* and Abdullah Kadri§
*Department of Electrical and Computer Engineering, Florida International University, Miami, FL, USA
§Qatar Mobility Innovations Center (QMIC), Doha, Qatar

***Abstract*—Zone-level occupancy tracking is a critical technology for smart buildings and can be used for applications such as building energy management, surveillance, and security. Existing occupancy tracking techniques typically require installation of large number of occupancy monitoring sensors inside a building as well as an established network. In this study, in order to achieve occupancy tracking, we consider the use of WiFi probe requests that are continuously transmitted from WiFi enabled smart devices. To this end, WiFi Pineapple equipment are used for *passively* capturing ambient probe requests from WiFi devices such as smart phones and tablets, where no connectivity to a WiFi network is required. This information is then used to localize users within coarsely defined occupancy zones, and subsequently obtain occupancy count within each zone at different time scales. Our numerical results using WiFi data collected at FIU over several days show that utilization of WiFi probe requests can be a viable solution for zone-level occupancy tracking in smart buildings.**

***Index Terms*—Localization, MAC randomization, occupancy, passive sniffing, probe requests, tracking, WiFi Pineapple.**

I. Introduction

Buildings are among the largest consumers of electricity in the United States: they account for 40% of primary energy consumption and 72% of electricity consumption [1]. An important portion of the electricity consumption of buildings is used for heating, ventilation, and air conditioning (HVAC). Occupancy tracking can help in achieving significant energy savings in smart buildings such as by dynamically scheduling HVAC activity based on real-time building occupancy levels at different areas [2]. Occupancy tracking can be implemented via video processing and camera systems or deployment of occupancy sensors throughout the building [3]. These options require installation of new equipment and are often costly to deploy. An alternative option is to use wireless signals of opportunity that uniquely match to building occupants. There are many solutions based on existing WiFi infrastructure, but they generally require a connection between user equipment and WiFi access points (APs) [3] [4]. In this paper, we use passive sniffing of WiFi probe requests for occupancy counting and tracking in smart buildings.

Probe requests are signals that are continuously broadcast from devices with WiFi technology, such as smartphones, laptops, and tablets [5]–[8]. When a WiFi client wants to get connected to a WiFi network, the first method is scanning for beacon frames, which are frames broadcast by WiFi routers to tell about their presence to WiFi clients [9]. The second method is sending probe requests, which also contains the unique MAC address of the device, as well its type, brand, manufacturer, and model. Since a WiFi client itself can initiate a connection to a WiFi router instead of waiting for a beacon frame from the router, use of probe requests is preferable. The probe requests are not encrypted, and can be passively captured and decoded with the help of wireless sniffers, without connecting to a particular network or transmitting any signal.

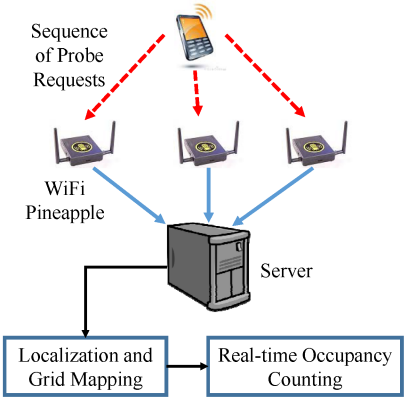

Fig. 1: Occupancy tracking using probe requests that are captured at multiple WiFi-PAs deployed in a building.

It is possible to capture the received signal strength (RSS) of probe requests using sniffers such as WiFi Pineapple (WiFi-PA) [10]. This information can then be used for occupancy monitoring inside the building. Probe requests are bursty in nature as they are broadcasted in the air in search of WiFi networks to get connected, to get a list of available networks, or to handover between WiFi APs. Frequent transmission of probe requests introduces an opportunity to track occupancy of building by passively sniffing and counting probe requests. To our best knowledge, there are no detailed studies in the literature that report efficiency of building occupancy tracking using WiFi probe requests.

In this paper, as summarized in Fig. 1, we use WiFi probe requests captured at various reference locations for occupancy monitoring in smart buildings. To this end, eight WiFi-PAs are deployed within the Florida International University (FIU) Engineering Center (EC) (see Fig. 2), and probe request data are passively sniffed over multiple days. The bursty nature of probe requests requires pre-processing of the data to make it ready for wireless location estimation. Subsequently, at every second, a linear least squares technique is used to obtain location estimates of each WiFi user, which are further refined using a weighted k-nearest neighbor (WKNN) location tracking algorithm. Location estimates are then aggregated into one of the eight occupancy zones inside the building. Our numerical results show that using probe requests, it is possible to effectively track the occupancy level information in different zones inside a building, which can be utilized in various smart building applications.

This paper is organized as follows. In Section II, a brief overview of existing literature about occupancy tracking and different uses of WiFi probe request are presented. Occupancy monitoring using probe requests are explained in Section III. Numerical and experimental results are provided in Section IV. Finally, concluding remarks and future prospects

978-1-5090-0448-5/16/$31.00 ©2016 IEEE

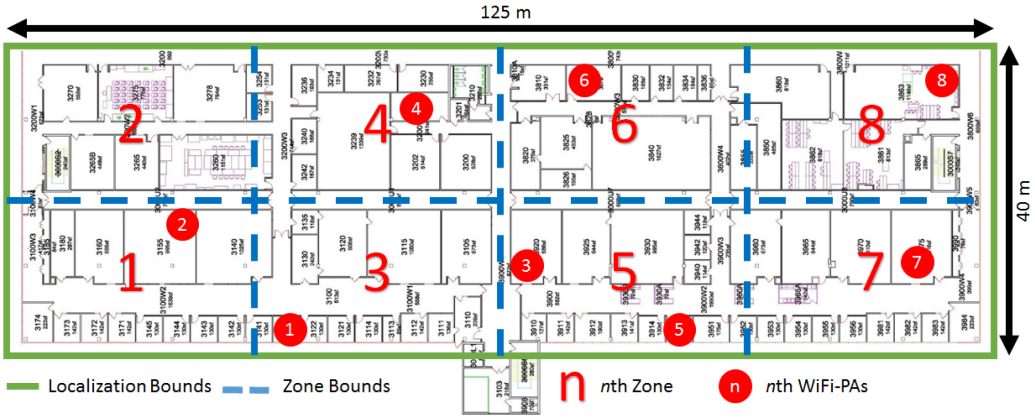

Fig. 2: Map of FIU Engineering Center 3rd Floor.

are summarized in Section VI.

II. Probe Requests: A Literature Review

Use of probe requests for location analytics and tracking, as well as privacy and security issues due to use of probe requests, have been recently gaining more interest in the literature. In this section, we will provide a brief survey of recent related work, which are classified and summarized in Table I.

A. Location Analytics, Search and Rescue

A common use case of probe request information is for location analytics of shoppers. In order to improve marketing, business teams can use probe request information to learn how frequently and when certain shoppers visit a store. Such information can be used for advertisement purposes, and to estimate required number of personnel for peak hours. For example in [11], Cisco Meraki APs are used as sniffers to capture probe requests from users. This data is then sent into the cloud, where a database is generated that involves location analytics and patterns of shoppers. To maintain privacy, only a hashed version of the MAC address is stored in the database. Other related work in the literature include [8], [9], which report data mining techniques to extract meaningful information from a database of probe requests captured from mobile users. Euclid Analytics [12] and Libelium [13] are other products that can be used for similar purposes.

In [14], it is shown that use of probe request information can be utilized in search and rescue operations. In particular, a WiFi-equipped drone actively broadcasts request-to-send (RTS) frames (100 per second) to trigger transmission of probe requests from a victim WiFi device. This information is then used to coarsely estimate the location of the WiFi device. Our work is different from the above studies in the sense that we capture the probe request for occupancy monitoring purposes. Privacy is maintained in the sense that the addresses are anonymized as in [11].

B. Trajectory Tracking

Probe request information have been used for second-by-second detection of a moving device (walking, driving) in [7] for outdoor environments. Subsequently, trajectory estimation of the mobile device is obtained using the Viterbi algorithm. Triangulation of mobile user locations by jointly using probe requests at multiple reference positions is not considered. Other recent work that study trajectory tracking using probe request information include [15]–[17].

Category	References
Location Analytics and Statistics	[8], [9], [11]–[14]
Trajectory Tracking	[7], [15]–[17]
Privacy and Security	[5], [6], [18]–[22]

TABLE I: Recent literature related to WiFi probe requests.

However, typically outdoor locations are considered, and triangulation and zone-level localization of a mobile device's position, as targeted by our work, is not explicitly studied.

C. Privacy and Security

In [23], tracking of WiFi users by passively capturing probe requests using WiFi sensors deployed at various locations is presented. The mobile user is assumed to be within the zone of its strongest WiFi sensor, and as opposed to our work, no explicit triangulation techniques are considered to fuse information from multiple WiFi sensors. Various tests were carried out using Android and Apple phones/tablets. Results show that more probe requests are sent when the device is in active mode (screen on) and not connected to any network, which can be used for tracking individual users based on RSS information. Even though companies such as Libelium [13] and Euclid Analytics [12] claim that privacy preserving techniques are possible, [23] shows that de-anonymization of a protected dataset is possible using different attacks. In another work [6], various privacy-related threats due to use of WiFi probe requests and factors affecting transmit frequency of probe requests have been discussed. Tests were conducted by keeping the phones in many different configurations such as connected mode and airplane mode. The maximum probing frequency is observed to happen when a device attempts to connect to a known network in its area. It is also shown that for a commercially deployed MAC randomization mechanism, it is possible to re-identify anonymized probes.

In [19], an attack referred as KARMA is described, in which the attacker automatically sends beacon and probe response frames for every received probe request, to direct the clients to his own network. This allows full control of the data sent by connected clients over the attacker's network. Authors in [19] introduce a detection mechanism for KARMA attacks, in which directed probe requests are sent with random SSIDs, and based on received probe responses the attacker can be identified. In [20], it is shown that early stage of probe request WiFi attacks can be identified with help of neural networks. In [18], hash-based anonymization of MAC addresses captured from probe requests are shown to be defeatable, while [21], [22] report various privacy vulnerabilities of mobile devices due to use of probe requests.

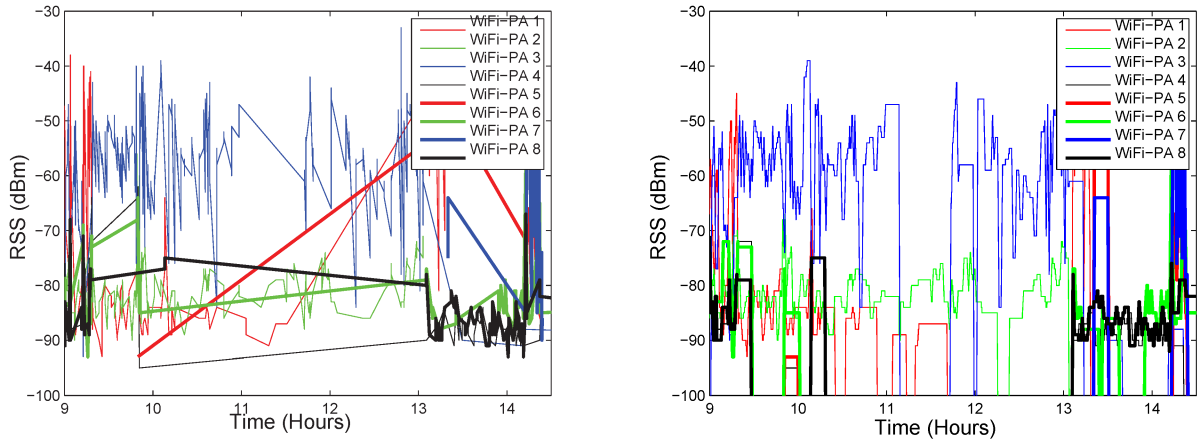

Fig. 3: Raw measurements (left) and processed data (right) of RSS.

III. Occupancy Monitoring Using Probe Requests

Use of probe requests for zone-level localization and tracking of building occupants is not a trivial task due to bursty availability of probe request information. In this section, we will first describe how to capture and pre-process probe requests at WiFi-PAs, and then will present a simple device localization and occupancy tracking technique using such data.

A. Capturing Probe Request Data with WiFi Pineapple

We conducted experiments in the Engineering Center of FIU to collect probe request information from building occupants. In the experiments, eight different WiFi-PA [10] are deployed at various locations at FIU EC 3rd Floor as shown in Fig. 2. WiFi-PA has dual integrated radios that comes with the Atheros AR9331 system on a chip (SoC), which includes a 400 MHz MIPS processor, 16 MB ROM and 64 MB RAM. All the WiFi-PA are accessed using the web interface or an in-built Unix machine called *BusyBox* [24]. We use the `tcpdump` sniffer to capture WiFi packets. As we are only interested in receiving the WiFi probe requests, all the other packets are filtered out and only WiFi probes are received. The data captured at WiFi-PA includes time stamps providing the time at which the data was captured, MAC address of the WiFi enabled device, and the signal strength of the WiFi device.

All the WiFi-PA are powered using a power adapter that is connected to an electric outlet. The scripts that we use to control each Pineapple can be executed via their terminal. They support boot modes which come with five user configurable switches that help in automated execution of multiple commands using the terminal. Using the web interface, we specify which script to run on which equipment, and as soon as the device is booted and switches points out to our script, it starts executing commands. One of the problems faced during our experimentation was that the internal clock time of the device keeps on changing every time it reboots which gives us wrong data capture timings. In order to address this problem, we forced each WiFi-PA to automatically connect to a WiFi network. Subsequently, using the *Network Time Protocol (NTP) Server*, the current date and time was synchronized with the time at the local network.

B. Database Server for Storing Probe Request Data

In order to obtain the data capture information from the deployed WiFi-PA, a Linux server has been set up. This makes it convenient to receive all the information at a centralized location. The data is stored in the SD card of the WiFi-PA for a day, and at the end of the day the captured data is sent to the server. Subsequently, each WiFi-PA starts capturing new data automatically, while also storing the back-up of the data in the SD card. The data is obtained in the packet capture format and can be viewed in *Wireshark* software [25], which is a packet analyzer that is used to convert the packets into comma separated variable (CSV) format that can be imported into MATLAB for further processing. The data is transferred from the the device to the server using Linux command `SCP`. In this command we enter the source address and the destination address we want to send the file. This also provides the time stamp of the file transfer, making it easy to search for particular packet capture file according to the date and time.

C. Pre-Processing Data for Location Estimation

The data captured and saved into server has to be pre-processed before using them in localization and tracking algorithm. First, the RSS data is resampled in time due to its bursty nature. For example, there may be several probe requests from the same user within few hundred milliseconds, followed by a silent period that may last several seconds. In order to have uniformly sampled RSS captures, we average the received RSS values within one second intervals.

After resampling the RSS measurements, the data needs to be interpolated for measurement intervals that do not have any RSS readings, and are also within close vicinity of other measurement intervals which have RSS readings. This is critical for localization, which requires at least four separate measurements from different WiFi-PAs (see Section III-D). The interpolation stage involves a sample-and-hold filter, which keeps the probe requests RSS value at a certain WiFi-PA for a fixed time window. If a new probe request is received, the RSS value is updated with the information obtained from that probe request. If no probe request is received within 300 seconds, the value of the RSS from that particular MAC address is labeled as unavailable.

In Fig. 3 raw measurements and processed data of RSS from 8 WiFi-PAs are shown, where the locations of WiFi-PAs are as reported in Fig. 2. The pre-processed data is more useful for our algorithm since it enables continuous availability of RSS information from bursty probe requests at multiple different WiFi-PAs, which improves localization accuracy. The cumulative distributions of RSS for each WiFi-PA and overall RSS are shown in Fig. 4. As it can be seen more than 70% of RSS values lie between -90 dBm and -70 dBm. This

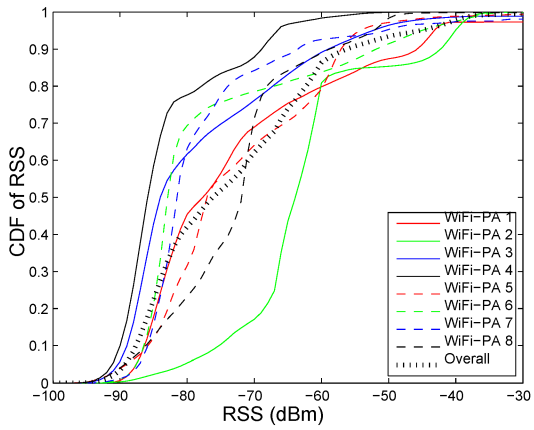

Fig. 4: CDF of RSS for different WiFi-PAs.

information is helpful about outlier detection. In this work, an RSS value smaller than -100 dBm or larger than -30 dBm is regarded as outlier and not considered for results.

D. Device Localization Using Probe Requests

After pre-processing of the data, localization techniques are used to get occupancy counts with respect to time for FIU EC 3rd floor. In [26], localization of WiFi APs are studied with unknown transmit power and path loss exponent (PLE). In our case we are localizing the devices with the help of WiFi-PAs, and transmission configurations of different WiFi equipment may have large variations [27]. Therefore, we use a similar technique to that of [26] for localizing users with unknown transmit powers.

The unknown position of a random user is denoted as (x_0, y_0) and the location of the ith WiFi-PA is denoted as (x_i, y_i), where $1 \leq i \leq k$. The RSS measurement in the ith position is denoted as r_i. The true distance between the user and the ith reference node is then given by:

$$d_i = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}, \tag{1}$$

and the path-loss model for the RSS is:

$$P_r = P_0 - 10n \log_{10}(d/l_0) + X_\sigma, \tag{2}$$

where P_r is the RSS, P_0 is signal strength at the reference distance l_0, n is the PLE which depends on the physical environment, and X_σ is the noise modeled by a Gaussian random variable with a standard deviation of σ and mean of zero. Using (2) the distance between transmitter and ith WiFi-PA is estimated as:

$$\hat{d}_i = 10^{(P_0 - r_i)/10n}. \tag{3}$$

Multilateration is the most known and used localization technique when distance estimates of at least three non-collinear reference positions are available. As it can be seen in (1), distance estimation requires the knowledge of transmit power and PLE. A linear approximation is used to overcome this situation in [26] as:

$$10^{(P_0 - r_i)/10n} \approx a_0 + a_1((P_0 - r_i)/10n), \tag{4}$$

where a_0 and a_1 are the linearization coefficients. It is shown in [26] that the linearization coefficients do not affect the localization accuracy. Subtracting all approximations of RSS from different WiFi-PAs, we can obtain:

$$\underbrace{\begin{bmatrix} -x_1^2 - y_1^2 + x_k^2 + y_k^2 \\ -x_2^2 - y_2^2 + x_k^2 + y_k^2 \\ \cdots \\ -x_{k-1}^2 - y_{k-1}^2 + x_k^2 + y_k^2 \end{bmatrix}}_{\mathbf{y}} =$$

$$\underbrace{\begin{bmatrix} -2x_1 + 2x_k & -2y_1 + 2y_k & \frac{a_1(r_1 - r_k)}{5} \\ -2x_2 + 2x_k & -2y_2 + 2y_k & \frac{a_1(r_2 - r_k)}{5} \\ & \cdots & \\ -2x_{k-1} + 2x_k & -2y_{k-1} + 2y_k & \frac{a_1(r_{k-1} - r_k)}{5} \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x_0 \\ y_0 \\ \frac{1}{n} \end{bmatrix}}_{\mathbf{x}} \tag{5}$$

where $k \geq 4$, and the k-th WiFi-PA is selected as reference for linearization. The final solution for $(x_0, y_0, \frac{1}{n})$ is given by:

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}. \tag{6}$$

E. Occupancy Tracking

Our final goal is real-time occupancy monitoring, by counting the number of users within coarsely defined occupancy zones in the area of interest. To this end, past location estimates of users should be considered collectively for accurate occupancy tracking. The very simple idea of using new location estimates solely to draw a trajectory leads to wasting the history of RSS and the previous estimates. There are also some models which take advantage of previous RSS measurements and location estimates in time.

In this paper, a weighted k-means algorithm is used for tracking users, in order to take advantage of their past location estimates. As a tracking algorithm, k-nearest neighbour is well-known and easy to implement as shown in [28]. Different than [28], we consider here that the algorithm weights the k-nearest neighbour measurements with respect to measurements' time difference to the last one, thus it can be called as time-weighted k-nearest neighbour measurements (WKNN). While using WKNN, a time threshold is also considered to increase the accuracy and reliability of tracking via discarding obsolete estimates. After using WKNN algorithm, the final estimate is mapped to the nearest zone point this time with distance threshold. If the resultant point is further away than a threshold to a zone's center, it can not be mapped to that cell.

The pseudocode for the WKNN tracking algorithm is shown in Fig. 5 where $\hat{\mathbf{x}}_k$ is the kth location estimate, t_i is the time difference between ith measurement and kth measurement, T is the threshold for obsoleteness of measurement, K is the set of last k measurements, w_i is the

1: **procedure** WKNN TRACKING
2: $\quad\hat{\mathbf{x}}_k = (\mathbf{H}_k^T\mathbf{H}_k)^{-1}\mathbf{H}_k^T\mathbf{y}_k$ ▷ k-th location estimate
3: $\quad$**for** $\forall t_i < T,\ i \in K$ **do**
4: $\quad\quad w_i = (T - t_i)/\sum_{i \in K}(T - t_i)$ ▷ Weight calculation
5: $\quad$**end**
6: $\quad\hat{\mathbf{u}}_k = \sum_{i \in K} w_i\hat{\mathbf{x}}_i / \sum_{i \in K} w_i$ ▷ Tracking result
7: $\quad\arg_j \min(D_j),\ \forall D_j < L,\ j \in M$
8: $\quad$**GridCell**$\leftarrow j$
9: **end procedure**

Fig. 5: Pseudocode for the proposed WKNN technique for tracking mobile devices.

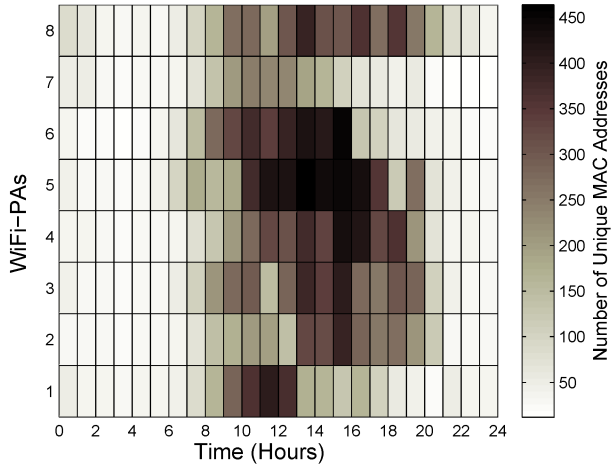

Fig. 6: Number of probe requests captured from each WiFi-PA.

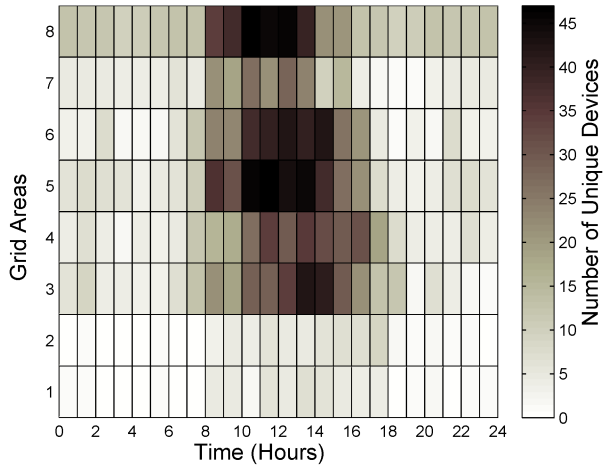

Fig. 7: Detected number of people in gridded areas.

weight of ith estimate, $\hat{\mathbf{u}}_k$ is updated sequence of location estimates after the tracking algorithm, D_j is the distance between tracking result and jth grid cell center, L is the distance threshold for mapping to a grid location, and M is the set of grid cells.

IV. Experimental Results

In this section, we will present several experimental results related to occupancy monitoring based on the collected and pre-processed probe request data.

A. Number of Unique MAC Addresses

One way to predict the occupancy count around a certain WiFi-PA is to count the number of probe requests received by that particular WiFi-PA. On the other hand, it might also be misleading since the number of overall probe requests can be increased solely by a particular WiFi device. For example, in our experiments we have seen that wireless printers may increase the number of probe requests heavily. For this reason, number of *unique* MAC addresses detected by a WiFi-PA gives a better hint about the number of WiFi devices and hence the occupancy of that region. To this end, in Fig. 6, number of unique WiFi probe requests captured at each WiFi-PA is

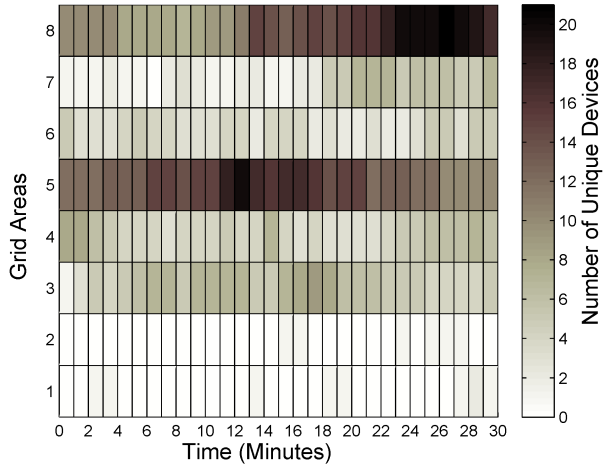

Fig. 8: Number of total people in gridded areas between 12 PM to 12:30 PM.

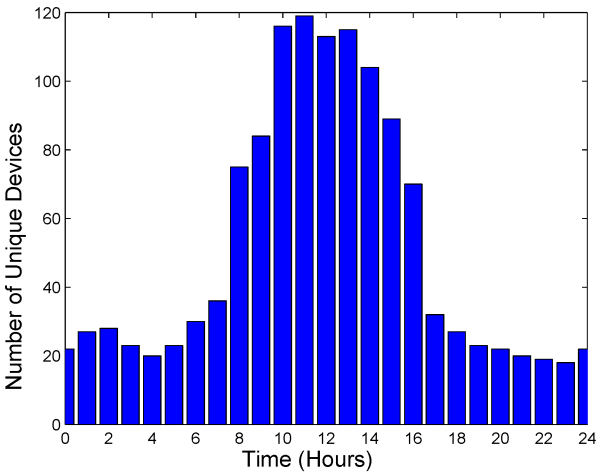

Fig. 9: Number of total people in gridded areas per hour.

presented as a function of time. It can be observed that there may be over 400 unique MAC addresses, captured within one hour at certain WiFi-PAs. Note that this number may also include devices that are in the vicinity of FIU EC building.

B. Occupancy Tracking

In order to get more accurate location information, WiFi users can be localized into one of the building zones, as described earlier in Section III-D. In Fig. 7, we explicitly triangulate users into one of the eight zones in Fig. 2, which requires RSS information to be captured at least by 4 WiFi-PAs. Results in Fig. 7 correspond to number of users observed in one hour time intervals within each zone. While the number of unique detected devices is lower compared to Fig. 6, Fig. 7 gives more reliable information about the occupancy pattern at different zones within the building. Fig. 8 provides occupancy pattern with one minute sampling interval between 12 PM to 12:30 PM, which can be used to extract fine-grained information for occupancy patterns.

C. Aggregate Occupancy in All Zones

In Fig. 9, total number of unique devices detected in gridded areas is given with respect to time, which shows that

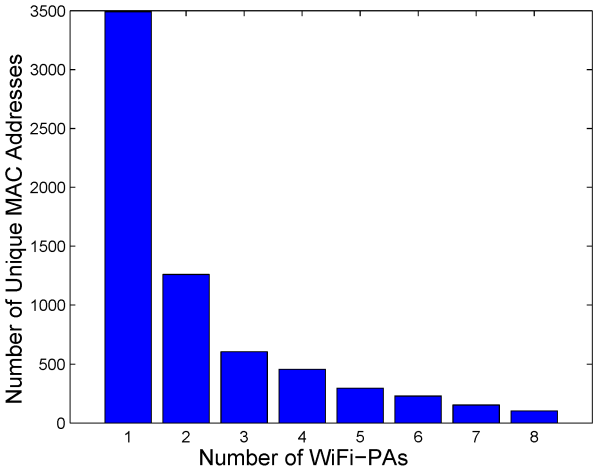

Fig. 10: Histogram of number of WiFi-PAs that see unique MAC addresses.

the peak hours of occupancy occurs between 10 AM and 2 PM. The number of detected devices drops to less than one fifth of peak hours after 4 PM. After midnight, there are on the order of 20 unique MAC addresses, which is aligned with the number of students working throughout the EC building after midnight. The difference between the number of devices detected and located can be explained with the help of Fig. 10. In the figure, the number of detected unique MAC addresses throughout the day is given with respect to number of WiFi-PAs that have detected them. It is clear that more than half of devices are detected by only a single WiFi-PA. As stated earlier, the device needs to be detected by at least four different WiFi-PA to be included in occupancy monitoring.

V. MAC Address Randomization

MAC address randomization feature has been recently introduced as a privacy preserving method in certain commercial smart phones, and may impact occupancy monitoring accuracy. In particular, if we would like to obtain occupancy count within a given time interval (e.g., one hour as in Fig. 7), if a smart phone changes its MAC address within that time period certain number of times, each different MAC address is counted as a separate user. Therefore, MAC address randomization may result in over-estimating the number of building occupants. In this section, we will provide an example experiment on how MAC address randomization works in iOS 8, leaving occupancy detection techniques with MAC address randomization as a future work.

The first 24 bits of the MAC address is considered as the Organizationally Unique Identifier (OUI), which differentiates the devices according to their manufacturers. The last 24 bits, on the other hand, are unique serial numbers assigned by manufacturers. The unique MAC address of devices can be easily captured and monitored from probe request messages since they are not encrypted. For example, owner of a network can restrict certain MAC addresses not to connect to a network. Having a fixed MAC address also introduces several security and privacy issues as discussed in Section II. In order to address such privacy concerns, recently, Apple introduced the iOS 8 operating system at Worldwide Developers Conference (WWDC) on September 17th, 2014. The most interesting feature in the new operating system was that MAC addresses will be randomized automatically making it difficult for an outsider to keep track of a user continuously.

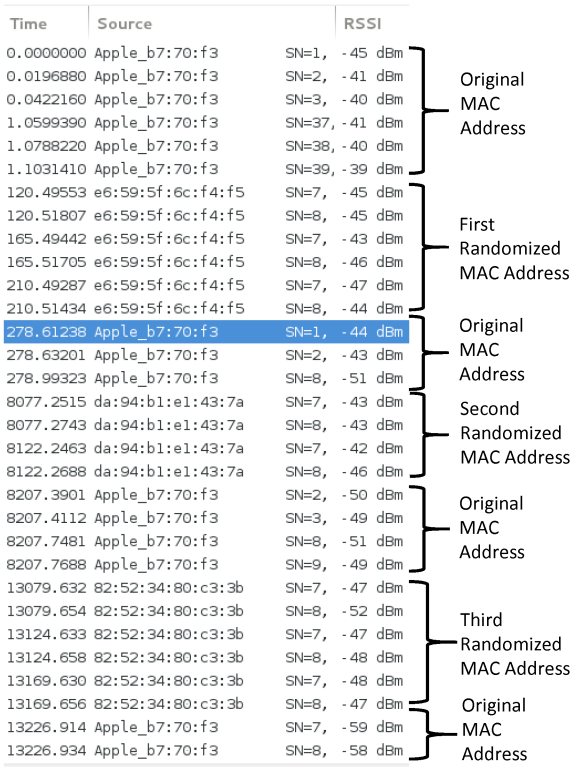

Fig. 11: Randomization of MAC address at a smart phone with iOS 8 operating system.

We used Wireshark to capture probe requests of an iOS 8.4 Apple phone and to evaluate the MAC address randomization feature of iOS 8. In order to have a randomized MAC address, the location services should be switched off, and the phone must first go into sleep mode while WiFi is switched on and not associated to a particular WiFi network. When the device goes into sleep mode, it takes around 150 seconds to send out randomized MACs, and it keeps on sending the same randomized MACs at equal intervals. If the device screen is used, and then the device goes into sleep mode again afterwards, the iO8 will be sending probes with randomized MAC, which is different than the randomized MAC in previous cycle [29]. This feature helps in changing the MAC address at every sleep cycle.

Experimental results for the considered iOS 8 device are shown Fig. 11. The real MAC address of the phone is `dc:86:d8:b7:70:f3` and first 24 bits is the OUI of Apple as identified by Wireshark. After the phone satisfies the conditions described earlier, MAC address gets first randomized into `e6:59:5f:6c:f4:f5`. When we bring the phone back from sleep mode it again starts emitting the real MAC, and when it goes into sleep mode a new MAC `da:94:ba:e1:43:7a` appears, which is different from the earlier randomized MAC. After repeating this cycle, the randomized MAC is observed to change again into another different MAC `82:52:34:80:c3:3b`. The RSS information for static and randomized MACs are also seen to be similar, which can be used as an additional information for detection of randomized MACs. When the device was connected to a network and then goes into the sleep mode, no randomization of MACs are found; instead, iOS will be probing for the previously connected network, as it gets disconnected with the network in sleep mode.

While the randomization of MAC addresses provides security to WiFi users, there are ways to identify randomized

MACs. For example, MAC addresses must be registered with IEEE standards association [30], and Wireshark automatically filters out MAC addresses which do not comply with this criteria. Even though MAC randomization feature is not provided by the Android operating system, it can be achieved in *rooted* Android devices by manually changing the MAC addresses, or using specific apps [31], which is rare compared to iOS 8 MAC randomization. In our study we determined 1464 MAC addresses which do not have any manufacturer's name, and hence they can be classified as randomized MAC address. We also identified 1639 iPhones based on the information collected from the probe requests. Since iOS 8 was the only operating system that implemented MAC address randomization at the time of the experiment, the 1464 randomized MAC addresses are expected to be mostly from a subset of the 1639 iPhones that have been identified.

VI. Conclusion and Future Work

In this work, occupancy tracking in smart buildings is studied by passively monitoring WiFi probe requests captured from smart phones and tablets. We use a linear least squares technique to estimate the location of a mobile device based on probe request information captured at multiple reference locations, which is then mapped into a building zone for coarse level occupancy tracking. Our results show that probe requests can be a viable solution for occupancy monitoring in future smart buildings, which can have application such as building energy management and surveillance. Our future work includes development of accurate occupancy tracking techniques that can operate effectively with observations at limited number of WiFi-PAs, and in the presence of MAC address randomization. We will also study privacy preservation techniques for occupancy tracking applications.

Acknowledgement

This research was supported in part by NSF under the grant numbers CNS-1446570 and ACI-1541108.

References

[1] V. Erickson, S. Achleitner, and A. Cerpa, "POEM: power-efficient occupancy-based energy management system," in *Proc. IEEE Information Processing in Sensor Networks (IPSN)*, Apr 2013, pp. 203–216.

[2] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng, "Occupancy-driven energy management for smart building automation," in *Proc. ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, 2010, pp. 1–6.

[3] K. Akkaya, I. Guvenc, R. Aygun, N. Pala, and A. Kadri, "IoT-based occupancy monitoring techniques for energy-efficient smart buildings," in *Proc. IEEE Wireless Commun. Networking Conference Workshops (WCNCW)*, Mar. 2015, pp. 58–63.

[4] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, and Y. Agarwal, "Sentinel: Occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings," in *Proc. ACM Conf. on Embedded Networked Sensor Systems*, 2013, pp. 17:1–17:14. [Online]. Available: http://doi.acm.org/10.1145/2517351.2517370

[5] L. Demir, "Wi-Fi tracking: what about privacy," Ph.D. dissertation, M2 SCCI Security, Cryptologyand Coding of Information-UFR IMAG, 2013.

[6] J. Freudiger, "How talkative is your mobile device?: An experimental study of Wi-Fi probe requests," in *Proc. ACM Conf. Security and Privacy in Wireless and Mobile Networks*, New York, NY, USA, 2015, pp. 8:1–8:6. [Online]. Available: http://doi.acm.org/10.1145/2766498.2766517

[7] A. B. M. Musa and J. Eriksson, "Tracking unmodified smartphones using Wi-fi monitors," in *Proc. ACM Conf. Embedded Network Sensor Systems*, ser. SenSys '12. New York, NY, USA: ACM, 2012, pp. 281–294. [Online]. Available: http://doi.acm.org/10.1145/2426656.2426685

[8] D. Namiot and M. Sneps-Sneppe, "On the analysis of statistics of mobile visitors," *Automatic Control and Computer Sciences*, vol. 48, no. 3, pp. 150–158, 2014.

[9] D. Namiot, "On mining mobile users by monitoring logs," in *Proc. Information Access in Smart Cities Workshop (i-ASC)*, Apr. 2014. [Online]. Available: http://dcs.gla.ac.uk/workshops/iASC2014/papers/iasc2014_namiot.pdf

[10] Hak5. (2013) Wi-Fi Pineapple Mark V. [Online]. Available: http://hakshop.myshopify.com/products/wifi-pineapple

[11] Cisco, "Location analytics," Mar. 2015. [Online]. Available: https://meraki.cisco.com/lib/pdf/meraki_whitepaper_cmx.pdf

[12] Euclid Analytics. [Online]. Available: http://euclidanalytics.com/

[13] Libelium. [Online]. Available: http://www.libelium.com/company/

[14] W. Wang, R. Joshi, A. Kulkarni, W. K. Leong, and B. Leong, "Feasibility study of mobile phone WiFi detection in aerial search and rescue operations," in *Proc. Asia-Pacific Workshop on Systems*, ser. APSys '13. New York, NY, USA: ACM, 2013, pp. 7:1–7:6. [Online]. Available: http://doi.acm.org/10.1145/2500727.2500729

[15] M. Handte, M. U. Iqbal, S. Wagner, W. Apolinarski, P. J. Marrón, E. M. M. Navarro, S. Martinez, S. I. Barthelemy, and M. G. Fernández, "Crowd density estimation for public transport vehicles," in *Proc. EDBT/ICDT Workshops*, 2014, pp. 315–322.

[16] D. Nix, "Analysis of methods for mobile device tracking," *Technical Report*, Oct. 2013. [Online]. Available: https://www.eyeqinsights.com/wp-content/uploads/2013/10/eyeQ-Mobile-Device-Tracking-Study-Nix.pdf

[17] Z. Xu, K. Sandrasegaran, X. Kong, X. Zhu, J. Zhao, B. Hu, and C.-C. Lin, "Pedestrain monitoring system using Wi-Fi technology and RSSI based localization," *Journal of Wireless & Mobile Networks*, vol. 5, no. 4, Aug. 2013.

[18] L. Demir, M. Cunche, and C. Lauradoux, "Analysing the privacy policies of Wi-Fi trackers," in *Proc. ACM Workshop on Physical Analytics*, 2014, pp. 39–44.

[19] T. Kropeit, "Don't trust open hotspots: Wi-Fi hacker detection and privacy protection via smartphone," *BS Thesis*, Mar. 2015. [Online]. Available: https://www.emsec.rub.de/media/attachments/files/2015/03/BA_Kropeit.pdf

[20] D. N. Ratnayake, H. B. Kazemian, and S. A. Yusuf, "Identification of probe request attacks in WLANs using neural networks," *Neural Computing and Applications*, vol. 25, no. 1, pp. 1–14, 2014.

[21] M. Cunche, M.-A. Kaafar, and R. Boreli, "Linking wireless devices using information contained in Wi-Fi probe requests," *Pervasive and Mobile Computing*, vol. 11, pp. 56–69, 2014.

[22] B. Bonné, P. Quax, and W. Lamotte, "Your mobile phone is a traitor!–raising awareness on ubiquitous privacy issues with SASQUATCH," *International Journal on Information Technologies Security*, no. 3, 2014. [Online]. Available: https://brambonne.com/docs/bonne14sasquatch.pdf

[23] L. Demir, "Wi-Fi tracking: what about privacy," MS Thesis, 2013. [Online]. Available: https://hal.inria.fr/hal-00859013

[24] Busybox. [Online]. Available: http://www.busybox.net/

[25] Wireshark. [Online]. Available: https://www.wireshark.org/

[26] J. Koo and H. Cha, "Localizing WiFi access points using signal strength," *IEEE Commun. Lett.*, vol. 15, no. 2, pp. 187–189, Feb. 2011.

[27] G. Lui, T. Gallagher, B. Li, A. Dempster, and C. Rizos, "Differences in RSSI readings made by different Wi-Fi chipsets: A limitation of WLAN localization," in *Proc. Int. Conf. Loc. and GNSS (ICL-GNSS)*, June 2011, pp. 53–57.

[28] J. Figueiras and S. Frattasi, *Mobile positioning and tracking: from conventional to cooperative techniques*. John Wiley & Sons, 2011.

[29] B. Misra. (2014, Sep.) iOS8 MAC Randomization. [Online]. Available: http://blog.airtightnetworks.com/ios8-mac-randomization-analyzed/

[30] IEEE, "MA-L PUBLIC LISTING," 2015. [Online]. Available: http://standards.ieee.org/develop/regauth/oui/public.html

[31] O. Abukmail. (2015, Sep.) WiFi mac address changer. [Online]. Available: https://play.google.com/store/apps/details?id=com.wireless.macchanger&hl=en