Indoor Crowd Estimation Scheme Using the Number of Wi-Fi Probe Requests under MAC Address Randomization

Yuki FURUYA\(^{(a)}\), Nonmember, Hiromu ASAHINA\(^{(b)}\), Member, Masashi YOSHIDA\(^{(c)}\), Student Member, and Iwao SASASE\(^{(d)}\), Fellow

SUMMARY As smartphones have become widespread in the past decade, Wi-Fi signal-based crowd estimation schemes are receiving increased attention. These estimation schemes count the number of unique MAC addresses in Wi-Fi signals, hereafter called probe requests (PRs), instead of counting the number of people. However, these estimation schemes have low accuracy of crowd estimation under MAC address randomization that replaces a unique MAC address with various dummy MAC addresses. To solve this problem, in this paper, we propose an indoor crowd estimation scheme using the number of PRs under MAC address randomization. The main idea of the proposed scheme is to leverage the fact that the number of PRs per a unit of time changes in proportion to the number of smartphones. Since a smartphone tends to send a constant number of PRs per a unit of time, the proposed scheme can estimate the accurate number of smartphones. Various experiment results show that the proposed scheme reduces estimation error by at most 75% compared to the conventional Wi-Fi signal-based crowd estimation scheme in an indoor environment.

**key words:** Wi-Fi probe request, MAC address randomization, crowd estimation, building automation, automatic people counting

1. Introduction

The ability to estimate the total number of people is important to improve the convenience of utilizing facilities [1]. For example, by counting people in queues, airports, restaurants and shopping malls attempt to improve their services, and also prepare for an effective evacuation guidance during times of emergency [2]. Most of the existing works have realized such estimation by leveraging either image recognition techniques [3] or Wi-Fi signals [4], [5]. Since Wi-Fi signal based approaches are less affected by light intensity or high congestion degree than image recognition based approaches, we focus on the Wi-Fi signal based approaches.

The Wi-Fi signal based approaches estimate the number of people inside the facility based on the fact that smartphones periodically send Wi-Fi control frame, called probe request (PR), with their unique MAC addresses [4]. Nowadays, since more and more people have Wi-Fi embedded smartphones, the Wi-Fi signal based approaches can easily estimate the number of people by counting the number of unique MAC addresses in PRs.

However, a single smartphone might have more than one MAC address when MAC address randomization is implemented in the smartphone [6]. To be more specific, an original MAC address is replaced by various dummy MAC addresses to prevent attackers from tracking locations of people [7]. This results in a larger number of unique MAC addresses than the actual number of people. In order to solve this problem, Matte et al. [5] proposed a scheme to estimate the number of people based on the fact that PRs which are sent from the same smartphone record the same values in their subfields. Since some subfields, such as the number of antennas, should be the same in every PR, that scheme can identify dummy MAC addresses which are sent from the same smartphone by using the subfields. Martin et al. [6] have made further improvement in terms of accuracy of derandomization by leveraging the fact that smartphones respond to RTS messages that record their original MAC addresses. has to perform brute-force search by repeatedly sending RTS with different MAC addresses, which can cause serious delay and interference to other communications. To our knowledge, only [5] has dealt with the MAC address randomization and thus is considered as the conventional scheme without causing such delay and interference and thus is considered the conventional scheme in this paper. However, since values of the subfields depend on the smartphone model, this scheme estimates a smaller number of people than the actual number of people when multiple people have the same smartphone model. Thus, to accurately estimate the number of people in a facility, it is necessary to estimate the number of same smartphone models.

In this paper, we propose indoor crowd estimation scheme using the number of PRs under MAC address randomization. The main idea of the proposed scheme is to leverage the fact that the number of PRs per a unit of time changes in proportion to the number of the same smartphone model. Since the number of PRs per a unit of time tends to be similar among the same smartphone models, the number of a specific smartphone model can be estimated from the number of PRs which are sent from the same smartphone model. We have clarified this fact by collecting datasets of PRs from 57 smartphones. Experimental evaluation results show that the proposed scheme improves estimation accuracy of the number of people by at most 75% compared to [5].

The following is the paper structure. Section 2 de-
FURUYA et al.: INDOOR CROWD ESTIMATION SCHEME USING THE NUMBER OF WI-FI PROBE REQUESTS UNDER MAC ADDRESS RANDOMIZATION

2. System Model

Figure 1 shows a model of an indoor crowd estimation system. We assume that the system is implemented in an area inside a facility, hereafter referred to as a target area. It is assumed that the target area is a line-of-sight environment, and up to 20 people stay in the target area for several minutes. This system consists of three elements: people, a wireless monitor, i.e., a Wi-Fi monitor, and a server. The people have their own smartphones, and it periodically sends PRs which record randomized MAC addresses. The Wi-Fi monitor continuously collects the PRs and sends them to a server. The server estimates the number of people with a crowd estimation scheme that is not affected by MAC address randomization.

3. Conventional Scheme

The main idea of the conventional scheme [5] is to utilize the trait that PRs record the same subfields in a header even if they record randomized MAC addresses. Since subfields have to record some immutable and unique values for performing wireless communication, such as the number of antennas, the conventional scheme can identify dummy MAC addresses corresponding to the same smartphone.

3.1 Algorithm of the Conventional Scheme

The algorithm of the conventional scheme consists of the following two steps: i) filtering PRs, and ii) estimating the number of people in a target area. When a Wi-Fi monitor receives a PR, it filters the PRs that are sent from a smartphone in the target area by selecting PRs of which Received Signal Strength Indicator (RSSI) exceeds a predetermined threshold. The threshold is experimentally determined by measuring minimum RSSI of PRs which are sent from smartphones in the target area. After this filtering, the algorithm estimates the number of people by using subfields of PRs. In particular, the algorithm creates lists of names and values of subfields as a FingerPrint (FP) of each smartphone. Finally, the algorithm counts the number of unique FP s of smartphones as the number of people in the target area.

3.2 Shortcoming of the Conventional Scheme

The conventional scheme tends to estimate a smaller number of people when multiple same model smartphones exist within a target area. In general, since same model smartphones use the same values in the subfields of their PRs, the number of unique FP s does not change even if the number of smartphones of the same model changes. This shortcoming might become a problem for improving services in a facility, since the most major model accounts for 11.73% of all smartphones as of 2019 in Japan.

Therefore, to improve the accuracy of the estimation, it is important to estimate the number of smartphones of the same model.

4. Proposed Scheme

In this paper, we propose an indoor crowd estimation scheme using the number of PRs under MAC address randomization. The main idea of the proposed scheme is to leverage the fact that the number of PRs with the same FP increases linearly with the number of the smartphones with the same model. We have collected a real-world dataset of PRs and clarified that a specific smartphone model tends to send a constant number of PRs per unit time. Based on this property, the number of smartphones of a specific model can be estimated by dividing the number of observed PRs by the number of PRs that the smartphones of a specific model send per unit of time. To realize this, a Wi-Fi monitor has to collect all PRs without omission. However, a Wi-Fi monitor may fail to collect PRs with low RSSI since smartphones send a bunch of PRs with various transmission power levels within a small time interval. In order to deal with this problem, the proposed scheme aggregates a bunch of PRs within a small time interval into a single group, hereafter called bursts, and uses the number of bursts for the estimation. Since smartphones send at least one PR with their maximum transmission power, the number of bursts is hardly changed even if a Wi-Fi monitor fails to collect some PRs.

In the following sections, we explain details of our dataset and proposed algorithm for crowd estimation.

4.1 Datasets

Our dataset is composed of bursts that are collected from 57 smartphones with 32 models. We use Wireshark, which is an open-source packet analyzer, to collect bursts [8]. We collect bursts for ten minutes by each model of a smartphone in an environment where there are no strong interference signals. Table 1 shows the average number of bursts per minute, hereafter referred to as the burst pattern, and its standard deviation of each model. As shown in Table 1, the standard deviation of each model is between 0 and 2.13. This result indicates that each smartphone model tends to send.
Table 1  Average and standard deviation of bursts.

| Device - OS Version | Ave. | Stdev. |
|---------------------|------|--------|
| Google Pixel 3 - 9.0 | 10.9 | 2.13   |
| Google Pixel 3 XL - 9.0 | 6.2 | 0.42   |
| iPhone 6S - 11.1.1   | 7.6 | 1.78   |
| iPhone 6S - 11.1.2   | 9.6 | 2.12   |
| iPhone 6S - 11.2.6   | 4.6 | 0.7    |
| SH-04H - 8.0.0       | 10.9| 2.08   |
| SH-04H - 8.0.0       | 13  | 0.82   |
| SO-01K - 8.0.0       | 7.5 | 0.85   |
| SO-01K - 8.0.0       | 7   | 0      |
| SO-02K - 8.0.0       | 7.2 | 0.63   |
| SO-03K - 8.0.0       | 7.3 | 0.48   |

Table 2  Average number of bursts per minute for different connection states.

| Connection State | iPhoneXR Ave. | iPhoneXR Stdev. | iPhone7 Ave. | iPhone7 Stdev. |
|------------------|---------------|-----------------|--------------|---------------|
| Not connected to AP | 2.16          | 1.35            | 2.34         | 0.75          |
| Connected to AP   | 0.04          | 0.02            | 0.40         | 0             |

Fig. 2  The indoor environment for the experiments in a realistic situation.

Table 3  Average and standard deviation of bursts in a realistic situation.

| User  | Device - OS Ver. | Time     | Average | Stdev. |
|-------|------------------|----------|---------|--------|
| User 1| iPhone7 - 14.2   | 0 - 10min.| 3.6     | 2.57   |
|       |                   | 10 - 20min.| 1.8     | 1.88   |
|       |                   | 20 - 30min.| 2.3     | 1.73   |
| User 2| iPhoneXR - 14.2  | 0 - 10min.| 1.9     | 1.51   |
|       |                   | 10 - 20min.| 1.2     | 1.53   |
|       |                   | 20 - 30min.| 3.8     | 3.09   |
| User 3| 602SO - 8.0.0    | 0 - 10min.| 0.4     | 0.24   |
|       |                   | 10 - 20min.| 0.5     | 0.45   |
|       |                   | 20 - 30min.| 0.4     | 0.24   |

Table 4  Average number of bursts per minute for different actions.

| Action               | iPhoneXR | iPhone7 |
|----------------------|----------|---------|
| Keep screen on       | 1.28     | 1.62    |
| Keep screen off      | -        | -       |
| Turn on screen       | 4.62     | 5.45    |
| Receive notification | 5.00     | 3.75    |
| Open Webpage         | 4.00     | 4.29    |

least since User 3 uses his/her smartphone less frequently than other users. In order to determine the actions that cause more bursts, we conducted an additional experiment. In this experiment, users continuously perform one of the following actions for ten minutes: i) keep the screen on, ii) keep the screen off, iii) turn on the screen, iv) receive notification and v) open web browser. Table 4 shows the average number of bursts per minute. It can be observed that turning on the screen, opening the web browser and receiving notifications cause more bursts. Therefore, the proposed scheme might overestimate the number of people in a situation where people frequently perform such actions. However, in general, the notification is intermittent and people might refrain from using their smartphone in the places where they meet another person, such as coffee shops, restaurants and so on.

From the above, we argue that the proposed scheme is effective in indoor environments where people do not frequently interact with their smartphone. Moreover, changes of the PR pattern might not cause a large estimation error in the proposed scheme directly. Since the proposed scheme estimates the number of people by dividing the observed number of PRs by the number of PRs recorded in the server, the
changes less than the number of PRs recorded in the server are rounded down.

4.2 Algorithm

The proposed algorithm estimates the number of people from the PRs every \( M \) minutes. Figure 3 shows a flowchart of the proposed algorithm. The algorithm consists of the following three steps: i) creating groups of PRs for the same smartphone model, ii) converting PRs to bursts, and iii) estimating the number of people by counting the number of smartphones with the same model in each group. Firstly, the algorithm creates groups of PRs for the same smartphone model. To be more specific, FPs of PRs are created by the conventional scheme, and the PRs are grouped by FPs. Secondly, the algorithm converts PRs in each group into bursts based on their time stamp. By analyzing bursts in our dataset, we have found that every time interval of a burst is below 0.5 (sec). Based on this fact, the algorithm merges a series of PRs into a single burst if the difference of arrival times among them is less than 0.5 (sec). Finally, the algorithm estimates the number of people by counting the number of smartphones with the same model in each group. In particular, the algorithm first accesses to the server to get the average number of bursts that the smartphone model sends per \( M \) minutes. Then, the algorithm divides the number of bursts in a group with the number of bursts per \( M \) minutes. When smartphones with unknown burst patterns are mixed in, i.e., the server does not have a corresponding FP of the observed bursts, the proposed scheme counts the number of such smartphones as 1. By performing the procedures above for all groups, the number of smartphones in each group can be estimated. Finally, the algorithm outputs the total number of smartphones in all groups as an estimation result of the number of people.

5. Evaluation

In order to show effectiveness of the proposed scheme, we implement crowd estimation scheme in an indoor environment, i.e. target area. In particular, we use Wireshark as a Wi-Fi monitor, and MacBook Pro (13-inch, 2017, Two Thunderbolt 3 ports) as a server. The numbers of smartphones are 7, 14, and 21. Table 5 shows the set of smartphones used in the experiment. In order to confirm both the advantage and disadvantage of the proposed scheme, we chose smartphones for the experiments so that the following three types of smartphone models are included: i) the models that have the single unique FP, ii) the models that have multiple FPs, and iii) the models that have the FPs common with other models. Since the proposed scheme assumes all smartphones are the model described in i), the models described in ii) and iii) can be causes of errors. If these errors cancel out the effectiveness of the proposed scheme, the performance of the proposed scheme should be the same as that of the conventional scheme. In each case, an experiment is repeated 4 times with different sets of smartphones, and average values of the estimation are used as estimation results. The predetermined threshold of RSSI for filtering outside PRs is \(-70\)dB and \( M = 1 \). In the experiment, smartphones and the Wi-Fi monitor use IEEE 802.11n. The Wi-Fi monitor observes PRs on channel 1 (2.412GHz). During the experiment, the smartphones are not associated with any APs, since smartphones associated with an AP do not use MAC address randomization.

5.1 Average Errors of Crowd Estimation

In this section, estimation accuracy of the proposed scheme and that of the conventional scheme are presented. Figure 4 shows average estimation errors from the ground truth, i.e., the actual number of smartphones, versus the number of smartphones. The error bars indicate the minimum and maximum errors. In Fig. 4, we can observe that average errors of the proposed scheme are lower than those of the conventional scheme. In particular, the average error of the proposed scheme among all experiments is 1.58, and that of the conventional scheme is 6.41. In other words, the proposed
scheme successfully reduces the estimation error by 75% compared to the conventional scheme. The reason behind this is that the results of the proposed scheme reflect the actual number of smartphones while those of the conventional scheme reflect only the number of unique FPs, i.e., the number of smartphone models. In fact, the errors in the conventional schemes are almost equal to the difference between the number of unique FPs and the actual number of smartphones. In this experiment, the average numbers of models of smartphones are 3, 7.5, and 13 while the actual numbers of smartphones are 7, 14, and 21 respectively. Thus, the differences between the number of models of smartphones and ground truth are 3, 7.5, 13. These differences are almost the same as average errors of the conventional scheme.

On the other hand, the proposed scheme also has estimation errors at most 5.98 when the number of smartphones is 14. By analyzing our dataset, we have clarified that this error is caused by the common FP among several different smartphone models. Due to such a common FP, the proposed scheme might not correctly estimate the number of smartphones with the same FP. In order to clarify the reason for the error of the proposed scheme, we will analyze the observed PRs in the next section.

### 5.2 FPs of Models of Smartphones

Table 6 shows correspondence between models of smartphones and FPs. Particularly, columns, rows, and each number in Table 6 show smartphone models, identifiers of FPs, and the number of observed PRs, respectively. Although almost all models of smartphones have unique FPs, several smartphone models have common FPs. For example, FPs of iPhone6S, Nova lite 2, and HMV32 are common with that of iPhone7, P10, and Mate10Pro, respectively. Since the proposed scheme relies on the unique FPs to get the unique fixed numbers of bursts, the proposed scheme cannot determine the appropriate number of bursts per a unit time of these models. Due to this reason, the estimation error of the proposed scheme slightly increases when a smartphone model has a common FP among other models. It is also observed that iOS tends to have common FPs among them in comparison with Android. One reason behind this might be the diversity of their specifications. iOS devices are made by a single company, i.e., Apple, and have similar specifications. In contrast, Android devices are characterized by their diversity. According to [5], PRs include data called “Tagged Parameters (TPs)” that are optional fields and the set of TPs can therefore vary from one device to another, depending on the configuration and capabilities of a device. We noted that, in general, the Android smartphones have unique TP and further investigated the reason behind this. Figure 5 shows examples of TPs of two different Android smartphones which are analyzed by a packet capture tool.
Wireshark. It can be seen that “Extended Capabilities” and “Vendor Specific” are different between two PRs. The reason behind this can be that each smartphone may use a different Wi-Fi chip depending on the price, and the relationship between smartphone vendors and Wi-Fi chip vendors. Since tags of the “vendor specifications” are determined by the vendor, the values are likely to differ with different vendors. Therefore, the sequence of IDs of the tags in the “vendor specifications” part is likely to be unique. Based on the above, we argue that it is reasonable to assume enough variations in FPs of Android smartphones, which have 116 vendors as of 2021 [10]. This also implies that the proposed scheme might have a limitation that it can estimate accurate number of people only in the case where almost all OS of smartphones are Android. However, since the world market share of Android in 2020 March is 73.0%, we argue that the influence of the common FP of iOS on the estimation is not a serious problem in real environment.

5.3 Influence of Unknown FP

The proposed scheme cannot estimate the number of smartphones with an unknown FP and thus the number of smartphones with an unknown FP is always counted as 1 even if there are several smartphones with the same FP. In order to evaluate the impact of these errors, we added an additional experiment results shown in Fig. 6 where the average errors from ground truth as a function of the ratio of unknown burst patterns can be found. It can be seen that although the estimation error increases as the unknown FP increases, the errors of the proposed scheme are smaller than those of the conventional scheme. This is because the conventional scheme always counts the number of smartphones with the same FP as 1. Moreover, the errors can be reduced by adding unknown FPs to the server.

5.4 Discussion

Although the proposed scheme assumes that the burst patterns are constant, the burst patterns are expected to change rapidly as time goes by. However, our proposed scheme will be effective by adding FPs of PRs which are not recorded in the server unless the randomization algorithm is changed such that the burst patterns dynamically change. In the future, when smartphones that send bursts at random times appear, the proposed scheme will not be able to estimate the exact number of people. Enhancements to overcome such a problem should be considered as future work. However, we argue that the burst patterns do not change rapidly based on the following two reasons. Firstly, in general, the only a limited number of Android smartphones can be upgraded to newer versions due to the fact that the software updates are managed by individual vendors. Therefore, even if a software update that includes the improved MAC address randomization is released, most of smartphones do not get that update immediately. According to a recent report of the share of android versions worldwide[11], the only 12.44% of Android devices use the latest version, i.e., Android 11, eight months after its release. Moreover, 3.1% of Android smartphones still use the oldest version Android 5.1. These facts imply that the diffusion of the new version is slow and the situation will not change suddenly. Secondly, the randomization mechanism may remain the same even if the smartphone version is upgraded since the randomization mechanism might be changed by the vendor. For example, SC-02J does not use the randomization even if its Android version is 9.0, i.e., a version released after the MAC address randomization has been implemented. Based on the above, we argue that the proposed scheme can be effective in estimating the number of people after the versions of operating systems referred to in this paper are becoming outdated.
6. Conclusion

We have proposed an indoor crowd estimation scheme using the number of PRs under MAC address randomization. To improve the estimation accuracy of the Wi-Fi signal-based crowd estimation under the MAC address randomization, we leverage the fact that the number of bursts per a unit time changes in proportion to the number of the same smartphone model. Evaluation results show that the proposed scheme reduces estimation error by at most 75% compared with the conventional scheme. We have also clarified that smartphones with similar specifications or ultra power-saving mode might degrade the estimation accuracy. In future extention, we will investigate other factors to improve the accuracy of crowd estimation.

References

[1] S. Depatla and Y. Mostofi, “Crowd counting through walls using wi,” 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp.1–10, 2018.
[2] H. Zou, Y. Zhou, J. Yang, W. Gu, L. Xie, and C. Spanos, "FreeCount: device-free crowd counting with commodity wi," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, pp.1–6, 2017.
[3] D. Figueria, M. Taiana, J.C. Nascimento, and A. Bernardino, “A window-based classifier for automatic video-based reidentification,” IEEE Trans. Syst., Man, Cybern., Syst., vol.46, no.12, pp.1736–1747, 2016.
[4] J. Wannenburg and R. Malekian, “Physical activity recognition from smartphone accelerometer data for user context awareness sensing,” IEEE Trans. Syst., Man, Cybern., Syst., vol.47, no.12, pp.3142–3149, 2017.
[5] M. Vanhoef, C. Matte, M. Cunche, L.S. Cardoso, and F. Piessens, “Why MAC address randomization is not enough: an analysis of Wi-Fi network discovery mechanisms,” ASIA CCS ’16, Xi’an., China, pp.413–424, May 2016.
[6] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E.C. Rye, and D. Brown, “A study of MAC address randomization in mobile devices and when it fails,” in Proceedings on Privacy Enhancing Technologies, vol.2017, no.4, pp.365–383, March 2017.
[7] J. Freudiger, “How talkative is your mobile device? an experimental study of Wi-Fi probe requests,” Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, WiSec ’15, New York, NY, USA, Association for Computing Machinery, 2015.
[8] Wireshark & Ethereal Network Protocol Analyzer Toolkit, Angela D ORebaugh, Gilbert Ramirez, Jay Beale, Joshua Wright, 2007.
[9] L. Oliveira, D. Schneider, J. De Souza, and W. Shen, "Mobile device detection through wifi probe request analysis," in IEEE Access, vol.7, pp.98579-98588, 2019, doi:10.1109/ACCESS.2019.2925406.
[10] https://www.gsmarena.com/makers.php3? All mobile phone brands [Internet].
[11] https://gs.statcounter.com/os-version-market-share/android Android Version Market Share Worldwide - April 2021 [Internet].