Characterizing probe request bursts to efficiently count Wi-Fi devices with randomized MACs

Abhishek Kumar Mishra, Mathieu Cunche

Characterizing probe request bursts to efficiently count Wi-Fi devices with randomized MACs

Abhishek Kumar Mishra
INSA-Lyon, Inria, CITI Lab., Lyon, France

Mathieu Cunche
INSA-Lyon, Inria, CITI Lab., Lyon, France

***Abstract*—In this paper, we show that counting the number of devices in a geographical zone is possible by passively capturing Wi-Fi probe-requests, even in the presence of randomized MAC addresses. We utilize a clustering-based approach and carefully characterize probe-request bursts with features that tend to be specific to a device. On three datasets with different capture configurations, we show that our methodology successfully counts the number of devices with a maximum error of 1 device.**

I. Introduction

The proliferation of Wi-Fi-connected devices enables diverse applications like user tracking and pedestrian flow estimation, yet raises privacy concerns, especially regarding anonymity and traceability. Modern Wi-Fi devices use active scans, transmitting probe-requests, which can be intercepted, posing privacy risks, addressed by measures like MAC address randomization. [1]–[3]

Our study demonstrates accurate inference of devices even in the presence of randomized MAC addresses through behavior analysis, leveraging diverse datasets. Analyzing publicly available probe-request frames allows the identification of the number of devices, as several features showcasing time, frame-content, and user-behavior-based information that tend to show a user-specific behavior.

II. Associating randomized MACs

In this section, we first look at Wi-Fi active scanning which results in emitting probe-request bursts before motivating the exploitation of burst-based metrics for associating the randomized MAC addresses. Finally, we proceed to define our proposed model and select features that could discriminate various devices.

A. Wi-Fi active scanning

Devices equipped with Wi-Fi capabilities employ active scanning (cf. Figure 1) to discover nearby wireless networks, sending out probe-request frames to explore accessible networks. When an access point (AP) detects a matching probe-request frame, it responds with a probe-response frame directly addressed to the requesting client, allowing the client to evaluate network options based on criteria like signal strength and security settings. To save energy, devices periodically broadcast probe-request frames, conducting multiple rounds of active scanning across available channels.

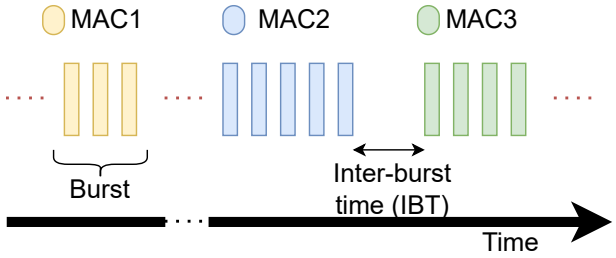

Fig. 1: Wi-Fi active scanning from a device using randomized MAC addresses.

We call each of the active scanning rounds which typically lasts less than a second, depending on factors like the number of known access points and channel availability, as a *burst*. Multiple probe-request bursts from a user can captured by deployed sniffers with each device maintaining consistent MAC addresses per burst. However, MAC randomization occurs, changing MAC addresses either in subsequent bursts or after a certain number of bursts. As shown in Figure 1, a device with address `MAC2` in a burst changes to another randomized address `MAC3` in the next burst. The time gap between two consecutive bursts is called as inter-burst time (IBT). We consider isolating and characterizing the individual bursts from the device as a first step to associate various randomized MAC addresses to their sending devices.

B. Characterizing probe-bursts

We consider various features (cf. Table I) that characterize the behaviour of probe-request bursts:

1. Time-based features: We select three such features.

- *The duration of the burst (T_b):* T_b measures the duration for which a single burst was observed at the receiving sniffer.
- *Mean IFS (μ^{IFS}):* μ^{IFS} denotes the average inter-frame space (IFS) for frames that are part of a single burst.

- *Size of the burst (S_b):* It states the number of frames in the considered burst.

TABLE I: Considered features.

$Metric$	$Feaure$	$Notation$
Burst-based	The duration of the burst	T_b
	Mean IFS	μ^{IFS}
	Size of the burst	S_b
Content-based	Number of present IE fields	N_{ie}
	Frame length	L_f
	Tag length	L_{tag}
	OUI	OUI
	Nature of MAC	R_f
Behaviour-based	Sojourn time of burst's MAC	T_{mac}

2. Content-based features: We have five frame content-based features.

- *Number of present IE fields (N_{ie}):* The probe-request frames do contain IE element fields which contain information about the device's capabilities and preferences. Out of around 256 specific elements that a smartphone could specifically advertise, in practice, many of them are not included. N_{ie} measures the number of non-empty IE fields for a random frame chosen from the burst.

- *Frame length (L_f):* The length of the frame.

- *Tag length (L_{tag}):* It refers to the length of the Tag field, which is used for various purposes such as carrying information elements.

- *OUI (OUI):* OUI (Organizationally Unique Identifier) refers to the first 24 bits of the MAC address. It uniquely identifies the organization or manufacturer responsible for the device's network interface.

- *Nature of MAC (R_f):* This feature states if MAC addresses in the burst are randomized or not.

3. Behavior-based features: We select two behavior-based features from the extracted bursts.

- *Sojourn time of burst's MAC (T_{mac}):* T_{mac} denotes the duration for which a particular MAC address is observed.

C. Model and Feature selection

Device counting by associating MAC addresses can be analyzed as a clustering problem. In this scenario, each probe-request burst provides valuable features that can be extracted and used as input for the clustering algorithm, while the algorithm's output associates the burst with a certain cluster denoting various MAC addresses belonging to a single device.

For efficiently using features from the set in Table I, we utilize DBSCAN, short for Density-Based Spatial Clustering of Applications with Noise. DBSCAN works by first identifying core samples, which are data points that have a minimum number of neighboring points within a specified radius. These core samples are then used to expand clusters by adding neighboring points that also have a minimum density of nearby points. Points that do not meet these criteria are classified as outliers. It is particularly effective for datasets characterized by clusters of comparable density, which is the case in clustering randomized MAC addresses.

III. Evaluation Methodology

Probe-requests from each device observed in the datasets are divided into separate bursts by identifying frame sequences with inter-frame durations exceeding 1 second. Only bursts containing multiple captured frames are considered, with the MAC address of a device remaining constant throughout a burst, serving as an identifier for the transmitting device. This facilitates the computation of burst-based features required for clustering. Feature sets from individual bursts (b_n) act as an input for DBSCAN.

DBSCAN has a critical `eps` parameter which is the maximum distance defining whether one sample is considered in the neighborhood of another, which is not a bound on the distances between points within a cluster. We find the optimal value of `eps` from a test dataset with a known number of devices sending probe-requests. We use the `scikit-learn` Python library[1], which provides the implementation of DBSCAN. We treat each non-clustered burst to be a new device in the sniffing zone that is possibly a passer-by.

We have three datasets that we utilize for evaluation of our work: i) `Capture A`: it was obtained by merging individual device captures. It contains the probe requests of 9 devices. ii) `Capture B`: This was obtained by merging individual captures. iii) `Capture C`: It was obtained by sniffing a group of devices inside the anechoic chamber.

TABLE II: Clustering Results

Metric	True devices	Device count
`Capture A`	9	10
`Capture B`	15	16
`Capture C`	22	23

IV. Results

Utilizing the dataset `Capture A`, we find the optimum value of `eps` to be 15. The results of our model are illustrated in Table II. We observe that for all possible capture settings our methodology captures the device count with an error of only one device. Additionally, in `Capture`

[1]https://scikit-learn.org/stable/index.html (version 1.3.2)

`A` and `Capture B`, we observe a high V-measure of around 94% and 80% respectively.

This attests to the effectiveness of our proposed WiFi device-counting methodology in various real-world deployment settings (with a known or unknown number of general population in a geographical zone, where passive sniffers are deployed).

REFERENCES

[1] Z. Koh, Y. Zhou, B. P. L. Lau, *et al.*, "Multiple-perspective clustering of passive Wi-Fi sensing trajectory data," *IEEE Transactions on Big Data*, 2020.

[2] B. Huang, G. Mao, Y. Qin, *et al.*, "Pedestrian flow estimation through passive WiFi sensing," *IEEE TMC*, 2021.

[3] A. K. Mishra, A. Carneiro Viana, N. Achir, *et al.*, "Public wireless packets anonymously hurt you," in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, 2021, pp. 649–652.