POLITECNICO DI TORINO
Repository ISTITUZIONALE

What WiFi Probe Requests can tell you

Original
What WiFi Probe Requests can tell you / Rusca, R., Sansoldo, F., Casetti, C., Giaccone, P.. - ELETTRONICO. - (2023), pp. 1086-1091. (IEEE Consumer Communications & Networking Conference (IEEE CCNC 2023) Las Vegas, NV (USA) 8–11 January 2023) [10.1109/CCNC51644.2023.10060447].

Availability:
This version is available at: 11583/2972106 since: 2022-10-05T16:16:09Z

Publisher:
IEEE

Published
DOI:10.1109/CCNC51644.2023.10060447

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

26 November 2026

What WiFi Probe Requests can tell you

Riccardo Rusca
Politecnico di Torino
Turin, Italy
riccardo.rusca@polito.it

Filippo Sansoldo
Politecnico di Torino
Turin, Italy
filippo.sansoldo@studenti.polito.it

Claudio Casetti
Politecnico di Torino
Turin, Italy
claudio.casetti@polito.it

Paolo Giaccone
Politecnico di Torino
Turin, Italy
paolo.giaccone@polito.it

***Abstract*—Everyday, as we go about our business in a city, we carry around several devices such as smartphones, tablets or even laptops, most of them with an active WiFi interface. This interface "leaks" wireless traces, or footprints, that can be used to identify the presence of people in certain areas. In particular, the analysis of devices' footprints allows the detection, tracking and monitoring of people in indoors and outdoors scenarios. In this paper, we focus on the probe request messages broadcasted by smart devices and we analyze the behaviour and the characteristics of these messages in different devices, coming from various vendors, with different operating systems and characteristics, also considering the user interaction with them. In particular, we provide a detailed picture of the adoption of MAC address randomization techniques, and on the variety of fields present within the probe request messages.**

***Index Terms*—Probe request, Passive sniffing, WiFi, People counting, MAC randomization, Raspberry Pi**

I. Introduction

In the recent years, several researchers and companies such as [1] have developed very low-cost hardware and software products, which, thanks to the analysis of broadcast device fingerprints and the power of machine learning algorithms, provide approximate counts of people in an area and can detect flows of passer-bys.

The starting point of all these studies are the so called "probe requests", they are messages defined by the IEEE 802.11 standard, which allows smart devices (e.g., smartphones, laptop, tablet, smartwatch, etc.) to request information to the access points (APs) in order to accelerate the process of connection to the WiFi network. These messages are sent periodically by the devices for searching known APs to connect or, if they are already connected to a WiFi network, to search an AP with an higher power strength and thus with higher performance. One of the big problems of using this kind of messages is the widely diverse behaviour of them. Each operating system (OS) and each device vendor usually customize the way of sending these packets over the network, as well as the frequency, but above all, the data contained within them.

The aim of this paper is to provide a comprehensive analysis on how often smart devices broadcast network probe requests, the frequency of bursts, and the data contained in them. Furthermore, we have analyzed how user interactions with the smartphone leads to the modification of the behavior of sending probe request messages by the device itself. To the best of our knowledge, only the authors of [2] have tried to answer the question of how "talkative" is a mobile device, but in the meanwhile, technology, OSs and smart devices have been completely redesigned. Also, the analysis was performed only on a couple of smartphones, and no tablets or laptops were considered.

The main contributions of our paper are:

- a detailed analysis of the behaviour of probe request messages, and the collection of information about frequency of bursts, number of packets per burst, main available parameters depending on the type of device (i.e., smartphone, tablet, and laptop), the vendor, the version of the operating system, and the device itself if used by the user;
- a detailed analysis of the behaviour of MAC address randomization, how it is applied, and how frequent the MAC address is randomized.
- a detailed analysis of the correlation between parameters of different probe request messages in a single device.

The rest of the paper is organized as follows. Sec. II presents an overview of the hardware used for the tests, as well as the used devices. The detailed description of the methodology followed is reported in Sec. III. The results obtained by the analysis of the captured files and the main takeaway messages are then provided in Sec. IV. A discussion of related work is in Sec. V and we draw our conclusions in Sec. VI.

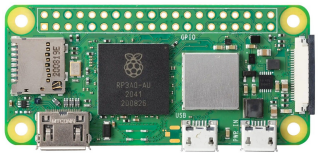
II. Hardware description

We provide an overview of the hardware used in the evaluation of probe request behavior, i.e., the sniffer and the devices under test.

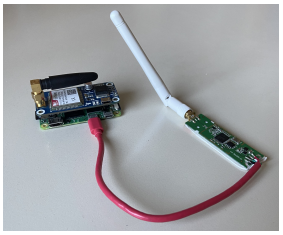
A. Sniffer

The tests were performed using a purpose-built sniffer, using off-the-shelf components, assembled by Dropper company [1] specifically to perform the test. More precisely, it is composed by a Single-Board-Computer (Raspberry Pi Zero 2), as shown in Fig 1a, equipped with an external WiFi card (chipset Atheros AR927) capturing in monitor mode the WiFi traffic. Furthermore, an interface for cellular connection (SIM7000E NB-IoT HAT) was added, to enable an SSH connection with a laptop, used as control center.

The Raspberry Pi Zero 2 runs as OS a Raspbian GNU/Linux 11 (bullseye) and the capture software running on it was TShark 3.4.10. Thanks for the adopted filters, we



(a) Single-Board-Computer Raspberry Pi, model Zero 2



(b) Final assembly

Fig. 1: Packet sniffer used to capture probe request messages

Type	Vendor	Model	OS	Year
Smart Phones	OnePlus	Nord 5G	Android 11.0	2021
	Samsung	Note 20 Ultra	Android 12.0	2020
	Apple	iPhone 11	iOS 15.0.1	2019
	Xiaomi	Redmi Note 8T	Android 10.0	2019
	Huawei	P9 Lite	Android 7.0	2016
	Apple	iPhone 6	iOS 12.5.5	2014
Tablet	Apple	iPad 8	iPadOS 14.8.1	2020
Laptops	Lenovo	ThinkPad X13 Gen1	Windows 11	2021
	Apple	MacBookAir M1	macOS 12.1	2020

TABLE I: Device list

captured just the probe requests and wrote the corresponding data into a pcap file, for later processing.

The final assembly of the above components is shown in Fig. 1b.

B. Tested Devices

The devices under test have been selected based on multiple factors in order to maximize the diversity in terms of manufacturer, OS version and year, and to cover typical devices used nowadays. Table I shows, for each device, the vendor, the model, the year and the version of the operating system it runs. For the iPhone, the "private Wi-Fi address" option was enabled, as by default. Laptops and the tablet were chosen to check if a notable difference between smartphones and other types of devices corresponds to a change in the behaviors of probe request transmissions and MAC address randomization algorithm.

III. Experimental methodology

In this section we want to focus on the methodology used to carry out the experiments and the subsequent data analysis. In detail, in Sec. III-B we describe all the steps followed for each devices in order to replicate exactly the same results for all the devices, while in Sec. III-C we describe how the captured files were analyzed, the filters used, and the main parameters of the probe request message we focused on.

A. Test location

The location chosen to carry out the tests is the La Mandria Regional Park. It is a very large park on the outskirts of Turin, in Italy, surrounding the famous Royal Palace of Venaria Reale. It is the perfect spot to execute tests in a controlled environment, thanks to its vast open spaces, and the low turnout of people, also thanks to the chosen time for the test, i.e, a cold morning in early March.

Fig. 2 shows two perspectives of the chosen location, in this large area there are 3 gazebos, used respectively as (1) control center, (2) location of a device for creating an hotspot for some test, and (3) location for the sniffer. The distance between (1) and (3) is about 90m, while the gazebo (2) is located halfway between the other two.

B. Test procedure

At the beginning of the whole experiment two capture sessions were performed, one before turning off all the tested devices, in order to make sure that the sniffer and Tshark were working correctly. The second one, lasting 10 minutes, was performed after turning off all the devices, in order to verify that the channel was free of any transmissions from nearby devices that could interfere with the captured data. The remoteness of the location guaranteed that there were none.

The experiment for each smartphone is subdivided into four phases: Locked, Awake, Active and Changing AP. Each experiment is performed in two main scenarios, the first one when the device is not connected to any AP and the second one when it is connected to an AP, created through a third smartphone with active hotspot and disabled WiFi interface. The Changing AP phase corresponds to the time the device is disconnected to the first AP, the one we called 'OPEN', and connected to a second AP, called 'OPEN2'.

The description of each phase is as follows:

- Locked: the device is locked with WiFi enabled. The phase lasts approximately between 3 and 8 minutes (depending on the time to detect a minimum number of probe request messages).
- Awake: the device screen is being tapped approximately each 30 seconds with WiFi enabled.
- Active: the device is unlocked and the user navigates through the apps and settings of the device.
- Changing AP: when the device is disconnected form the first AP and connected to a new one.

For what concerns the tablet and the PCs the tests were quicker and consisted in an active use of the device, in the two scenarios (i.e., connected / not connected to the AP).

C. Data analysis

The python library PyShark has been used to read the captured data, and create the results showed in Sec. IV. The PyShark library acts as a wrapper for TShark, allowing Python to parse packets using Wireshark's built-in dissectors.

Our work is motivated by finding a way to distinguish a specific device among the others. Inside the probe request message, some fields could be used to identify uniquely a device and thus lead to a fingerprint: particularly, inside the frame body, fields such as the SSID, HT capabilities, extended capabilities, are all amenable to identify a device. Notably, the



Fig. 2: Test location in La Mandria, Turin, Italy

SSID can be a wildcard to connect to any possible nearby AP. We also study the length of the probe requests, in terms of management frame carried by 802.11 (i.e., including only the corresponding fields and excluding the 802.11 MAC header fields).

The main core of the analysis starts with filtering out the packets corresponding to the traffic to control the sniffer behaviour. Then, a filter on a minimum RSSI is applied as a further safeguard to isolate the sensing from possible external devices placed out of the experiment range of capture, if any. After that, our scripts parse each field of the packets and create a data structure that allows us to evaluate the temporal evolution of different metrics like burst length, burst inter-arrival time, message length and RSSI.

Finally, we process the so-called IE (Information Elements) fields, as proposed by the authors of [3] to identity univocally a specific device. These specific features include the HT capabilities, the SSID of the AP to which the device wishes to connect, the extended capabilities and the vendor ID.

IV. Experimental results

We now present all the results obtained by our experiments and discuss our main findings. For the sake of space, only the graphs related to one device are described in detail. Additional graphs related to all the other devices and the raw captures are available on a public GitHub repository [4].

A. iPhone 11 analysis

Figs. 3 and 4 show the results obtained by analyzing the captured probe requests from the Apple iPhone 11. Each histogram corresponds to an observation window of 30 seconds. The colored bands in the graphs identify the different test phases. The yellow band refers to the Locked phase, the green band to the Awake and the purple band to the Active phase; the pink band indicates the change of the AP the device is connected to. For this reason the dotted vertical green line divides the part in which the device is not connected to an AP and when it is connected to one: i.e., before the dotted green line the device is not connected to an AP, while after it, the device is connected to an AP.

Fig. 3 shows six different plots. From the top down, the first plot shows the number of packets received per period of time. As expected, during the active phase we can see more

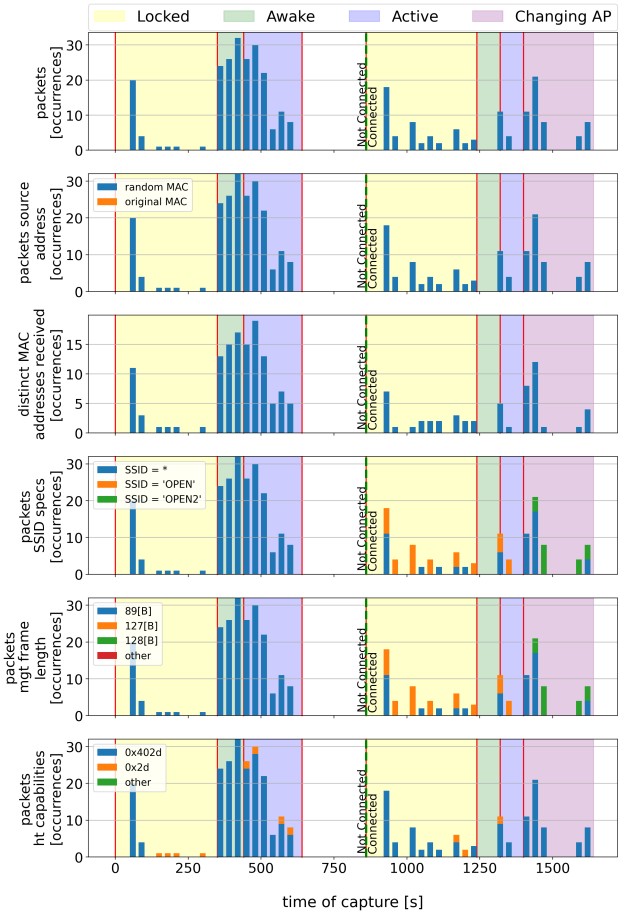


Fig. 3: Experimental results for iPhone 11 in function of the time, for different connection phases.

packets than in the locked phase and surprisingly also during the awake phase. The second plot shows the occurrences of classic and random source MAC addresses, this is particularly useful to understand if the device randomizes its MAC or not. In this case, the iPhone 11 is never sending its original MAC address. The third plot shows the overall number of distinct MAC addresses for the whole duration of the experiment.

When not connected to the AP, the number of distinct MAC addresses is 117 on the total of 213 packets, i.e., roughly half of the number of packets. This can be seen as the MAC address changes every two packets, coherently with the fact that in the following we will show that the packets are sent in burst of two, each burst with a distinct random MAC address. Interestingly, when connected to the AP, the number of distinct MAC addresses for each burst is reduced.

The fourth plot shows the occurrences of each AP SSID observed in the probes. Three values were observed: the wildcard SSID and the two preset SSIDs. After the connection, the device tends to search the AP to which it is connected and to specify its SSID in plaintext inside the probe. The fifth plot shows the occurrences of probe request lengths, in Bytes, in function of the time. In this case when the phone

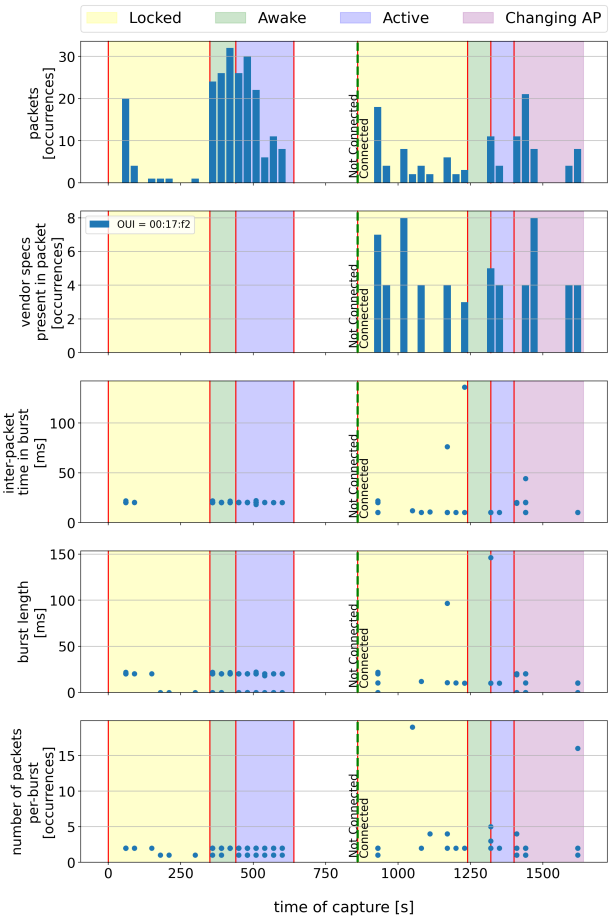


Fig. 4: Experimental results for iPhone 11 highlighting the bursts and the vendor field as a function of time.

is connected to an AP the length of the probe request frame increases, since the message now includes the SSID and some additional tags to specify the vendor. The last diagram shows the HT capabilities values and their frequencies in the packets. We can state that most of the time they are constant over the time.

Fig. 4 shows other details emerged during the analysis. In particular, the first plot is the same as in Fig. 3, and is left as a temporal reference. The second plot displays the packets occurrences where the vendor ID (OUI=00:17:F2) is specified and it clearly shows that the vendor ID starts to appear after the device is connected to an AP, even if the MAC address continues to be randomized. The last three plots report the interarrival time between packets within the same burst (equal to 20ms when not connected), the burst duration and the number of packets for each burst. Before connecting to the AP, all the bursts last 1 or 2 packets, each with a different MAC address. Thus we can conclude that MAC randomization occurs at burst level and not at packet level.

We also studied the temporal evolution of the sequence numbers and found out that they increase by one or few units within the same burst (i.e., with the same random MAC

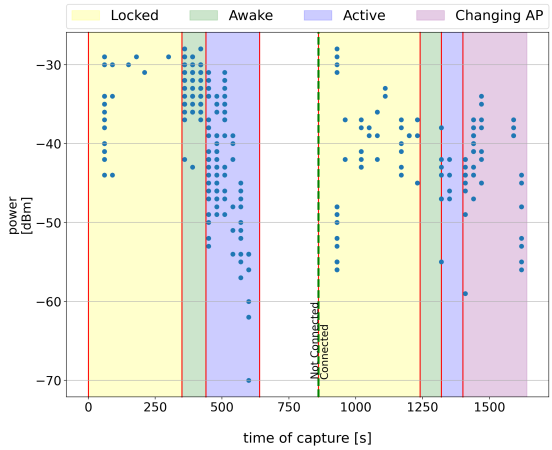


Fig. 5: Experimental results for iPhone 11 referring to the received power in function of the time.

address), whereas the initial sequence number of a new burst is chosen randomly or without a clear rule. This is again a clear countermeasure adopted by iPhone devices against fingerprinting.

1) Power analysis: For what concerns the power at which packets are received by the sniffer, it can be stated that there is no clear correlation with the phases of the test. Indeed, as can be seen in Fig. 5 the RSSI changes over time, but this is mostly due to small movements of the experimenter when the phone is active. Indeed, during the Locked phase, the device was kept still in the same place, whereas during the Active phase the user interacted with the screen with his fingerprints, changed the propagation conditions. In general, from our experimental evidence the RSSI is very erratic, even if the smartphone was in a fixed position, and cannot be used to identify univocally a device and/or the corresponding phase.

B. Analysis and comparison of the other devices

We performed the same detailed analysis for the experimental results obtained for all the other devices, reported in [4]. For the sake of space, we report here just the final conclusions. Tables II and III compare the features of the different devices, smartphones, tablet, laptops, in the following categories:

- MAC address randomization;
- Burst characteristics;
- Sequence number;
- High Throughput (HT) Capabilities;
- Management (MGT) Frame;
- SSID Field.

As one can see, nearly every device exhibits its own specific behavior in each of the above categories, further complicating the task of finding guidelines in an attempt to quantify the number of distinct devices (hence of people carrying them) in an area. In the next section, we will summarize our findings.

C. Experimental findings

In general, as main takeaway message, each device exhibits its own peculiar behavior.

	OnePlus Nord 5G (2021)	Samsung Note 20 Ultra (2020)	Apple iPhone 11 (2019)	Xiaomi Redmi Note 8T (2019)	Huawei P9 Lite (2016)	Apple iPhone 6 (2014)
MAC address Randomization	randomized at each burst	unconnected: changes at each burst; connected: constant random MAC address	unconnected: changes at each burst; connected: constant random MAC address	unconnected: changes at each burst; connected: same random address for each AP	always original, not randomized MAC address	unconnected: changes at each burst; connected: original, not randomized MAC address
Burst length	about 3/5 packets	unconnected: about 3/5 packets; connected: large bursts with same MAC address	unconnected: 2,3 packets; connected: longer	unconnected: 3,4 packets; connected: longer	MAC randomization not active	about 3/5 packets
Sequence Number	increases inside the burst, random starting number in different bursts	unconnected: changes randomly; connected: increases constantly	unconnected: changes randomly; connected: increases (almost) constantly	unconnected: changes randomly; connected: increases (almost) constantly	increases constantly	increases constantly
HT Capabilities	same text in each packet	same text in each packet	mostly same text in each packet	same text in each packet	changes depending on AP	same text in each packet
Management (MGT) Frame	unconnected: variable frame lengths; connected: constant length	unconnected: variable frame lengths; connected: constant length	unconnected: variable frame lengths; connected: length depends on associated AP	unconnected: variable frame lengths; connected: length depends on associated AP	frame length constant except for some packets when connecting	unconnected: variable frame lengths; connected: length depends on associated AP
SSID Field	never specified	SSID in plaintext in some packets; in others, strings not traceable to any SSID name used in the experiment	SSID in plaintext when connected; wildcard used to look for other APs	unconnected: wildcard; connected: SSID in plaintext; wildcard used to look for other APs	connected: SSID in plaintext; unconnected not in plaintext	SSID in plaintext when connected

TABLE II: Smartphones behaviour comparison

	Apple iPad 8 (2020)	Lenovo ThinkPad X13 Gen1 (2021)	Apple MacBookAir M1 (2020)
MAC address randomization	randomized at each burst	none	unconnected: changes at each burst; connected: original MAC address
Burst length	2/4 packets	MAC randomization not active	2/4 packets
Sequence Number	always changes randomly	always changes randomly	always changes randomly
HT Capabilities	same text in each packet	same text in each packet	same text in each packet
Management (MGT) Frame	constant length, which changes only when connecting to AP	constant length when looking for APs, then changes when SSID field is specified	constant length when looking for APs, then changes when SSID field is specified
SSID Field	wildcard, except when connecting to AP	SSID specified a few times also when connected to AP	wildcard, except when connecting to AP

TABLE III: Tablet and laptop behaviour comparison

The number of probe requests varies across the tested devices: Xiaomi sent only few packets in the overall capture, compared to the other ones, especially OnePlus which sent the highest number of probes.

The process of MAC randomization is very different across all of our devices. Randomization is mostly performed on a per-burst basis and the burst length slightly increases passing from iOS to Android. The implementation of MAC randomization does not seem to be consistent in all models by the same manufacturer, as it changed from iPhone 6 and iPhone 11. There are instances, such as with the Huawei P9 Lite (2016) or the Lenovo ThinkPad, where MAC addresses are not not randomized at all, and the real MAC address appears in plaintext in every probe request.

Likewise, the SSID field is not very reliable as unique identifier, since devices do not trasmit it on a regular basis and, in the case of OnePlus, it is never transmitted.

The sequence number found in packet bursts is also a varying feature: it increases linearly in older phones, while in newer ones it increases in random, large increments, making difficult to identify uniquely each device.

Other features are more amenable to be considered as fingerprints. For example, the HT capabilities seem to remain consistent in many devices, and only iPhone 11 changes them

rarely, while Huawei P9 changes them whenever connected to different APs.

The probe request, when the device is connected to an AP, carries also information regarding the connected AP (e.g., VendorID), thus its size is typically larger than when not connected.

Tables II and III summarize, for each device, the details of all features and behaviors that could be used to for fingerprinting. The "connected/unconnected" label refer to the state of connection to the AP.

V. Related Work

In recent years, several different approaches have been used to address the problem of detecting and counting people using probe request messages, by analyzing the MAC addresses and other parameters within the messages. However, most of the proposed techniques are no longer applicable as the information encapsulated within the probe request are changing over the time, as many of the major smart devices vendors started to customize this values to enhance the privacy of the users.

The work in [3] proposes an approach to design a derandomizer for MAC addresses, considering the Information Element (IE) of the probe requests and finding correlations among the packets lengths. In such way, when two probes have the same IE lengths, they are considered as coming from the same device. Notwithstanding the high accuracy obtained by the authors (up to 91%), we proved in our experiment that they are not constant for all the probe requests emitted by the same device, as it is possible to see in the last plot of Fig. 4. Similarly, [2] studied the behaviour of WiFi probe requests and how they affected the privacy of users in 2015. Since years passed and the manufacturers as the OSs evolved their techniques to enhance user privacy, we noticed that the behaviour of sequence number is completely changed and nowadays it is completely unreliable, also the frequency of burst is increased a lot.

The work in [5] designed a system to derandomize the probe requests inside controlled environments (classrooms and laboratories) in order to estimate the number of people. Likewise, [6] tried to derandomize requests on a bus, focusing on the distinction between people inside and outside the vehicle and using an efficient algorithm called iABACUS, which exploit the sequence number of different frames for the randomization. Unfortunately, as observed in Sec. IV-A and shown in Table II, sequence number can be exploited only within the same burst and iABACUS approach would lead to count the bursts and lead to a huge overestimation of the present devices.

Finally, other studies tried to recover information from the probe requests: the mechanism described in [7] shows that devices also advertize the SSID of the AP they want to connect to. The studies were conducted on large scale and were aimed to analyze and discover the provenience of people in the crowd, taking the so called PNL (Preferred Network List) from the probes and linking the SSID to the corresponding IP and then its localization. Nowadays recent OSs deleted that sensitive information from the probe request fields, as we showed in Sec. IV, thus this approach is not valid anymore.

Similarly to our experimental tests, an open dataset has been made public in [8], which provides the pcap traces of capturing the probe requests sent by a large set of smartphones. Similarly to our work, different operating modes of the smartphones have been considered. Differently from our work, some of the captures were taken in an anechoic chamber; furthermore, [8] describes in details the adopted experimental methodology and does not analyze the collected data. We leave as future work to extend our analysis to these traces.

VI. Conclusion

Given the widespread adoption of mobile phones and other smart devices, the ability to track their presence through the data messages they broadcast (e.g., the WiFi beacons) can lead to the estimate of the number of people in a certain area. The applications of such a capability are appealing, ranging from security to the dimensioning of public services. This goal is however thwarted by the diversity of broadcasting strategies and message formats, notwithstanding the standardization efforts. In this paper we have tried to analyse the behavior of a broad range of device types and operating systems, finding that several strategies, such as MAC address randomization, can hinder the effort to quantify the number of devices. However, looking at other parameters such as length of beacon bursts or specific fields in the beacons themselves, one can design more holistic approaches to, at least approximately, detect individual device broadcasts.

References

[1] Dropper. [Online]. Available: https://www.dropper.ai/

[2] J. Freudiger, "How talkative is your mobile device? an experimental study of Wi-Fi probe requests," ser. ACM WiSec, New York, NY, USA, 2015.

[3] M. Uras, R. Cossu, E. Ferrara, O. Bagdasar, A. Liotta, and L. Atzori, "WiFi probes sniffing: an artificial intelligence based approach for MAC addresses de-randomization," in *IEEE CAMAD*, 2020.

[4] R. Rusca and F. Sansoldo. Experimental data about probe request sniffing. [Online]. Available: https://github.com/riccardo-rusca/ProbeRequestSniffing

[5] P. Galluzzi, E. Longo, A. E. C. Redondi, and M. Cesana, "Occupancy estimation using low-cost Wi-Fi sniffers," Tech. Rep., 2019. [Online]. Available: http://arxiv.org/abs/1905.06809

[6] M. Nitti, F. Pinna, L. Pintor, V. Pilloni, and B. Barabino, "iABACUS: A Wi-Fi-based automatic bus passenger counting system," *Energies*, vol. 13, no. 6, 2020.

[7] A. Di Luzio, A. Mei, and J. Stefa, "Mind your probes: De-anonymization of large crowds through smartphone WiFi probe requests," in *IEEE INFOCOM*, 2016.

[8] L. Pintor and L. Atzori, "A dataset of labelled device Wi-Fi probe requests for MAC address de-randomization," *Computer Networks*, vol. 205, 2022.