A WiFi-based method to count and locate pedestrians in urban traffic scenarios

Antonio Guillén-Pérez
Department of Information and Communication Technologies
Universidad Politécnica de Cartagena
Cartagena, Spain
antonio.guillen@edu.upct.es

María Dolores Cano Baños
Department of Information and Communication Technologies
Universidad Politécnica de Cartagena
Cartagena, Spain
mdolores.cano@upct.es

***Abstract*— Estimating the number of people in a given environment is an attractive tool with a wide range of applications. Urban environments are not an exception. Counting pedestrians and locate them properly in a road traffic scenario can facilitate the design of intelligent systems for traffic control that take into account more actors and not only vehicle-based optimizations. In this work, we present a new WiFi-based passive method to estimate the number of pedestrians in an urban traffic scenario formed by signaled intersections. Particularly, we are able i) to distinguish between pedestrians walking and pedestrians waiting in a pedestrian crossing and ii) to estimate the exact location where static pedestrians are waiting to cross. By doing so, the pedestrian factor could be included in intelligent control management systems for traffic optimization in real time. The performance analysis carried out shows that our method is able to achieve a significant level of accuracy while presenting an easy implementation.**

***Keywords*— counting people, pedestrians, WiFi, Urban Computing, Intelligent Transportation Systems**

I. Introduction

Counting or estimating people is a useful task for a wide range of services [1]: security, marketing and retail, transportation, urban planning, tourism, etc. Several methods have been proposed in the related literature to estimate the amount of people in indoor and outdoor scenarios. Most proposals can be classified into one of the following categories depending on the baseline procedure: image processing of capture video/images [2], sensor detectors [3], or radio frequency-based techniques such as Bluetooth or WiFi [4] [5] [6]. Given that i) most people carry out a mobile handset (e.g., a smartphone) that incorporates a WiFi module, ii) WiFi-enabled devices autonomously send broadcast Probe Request messages in a regular manner to discover available WiFi Access Points (AP), and iii) each Probe Request message contains a unique device identifier and/or information about the sending device, it is straightforward to think in a method that captures and processes those frames in order to distinguish devices, to identify their motion pattern, to perform tracking, etc. WiFi-based counting methods that do not require an association between the mobile device and the AP are called *passive* methods, whereas methods that required the establishment of an association among devices and AP are called *active* methods. Please note that along this paper we will use interchangeable the terms mobile device, smartphone, or mobile handset, to refer to a mobile communication device that includes a WiFi communication module.

Despite the simplicity of the idea, practice reveals that WiFi-based passive methods for people counting need to overcome some difficulties, in turn, influencing the accuracy of the system. To start with, not everybody carries a mobile handset, and if does, the WiFi may not be enabled (e.g., for security reasons or to save battery). Consequently, these people are present, but they would not be counted. On the contrary, there could be people with more than one active WiFi device, and thus counting as more than one person. Additionally, using a radio-based medium involves dealing with non-deterministic factors that affect the communication channel such as fading, noise, distortion, etc., hence representing a negative effect on the system performance. Also, APs could be out of reach for a short period of time because of a rebooting, thus partially losing some information about the WiFi devices. Similarly, Received Signal Strength Indicator (RSSI) measurements are very variable in terms of the specific characteristics of the AP that is capturing the Probe Request frames and the mobile device manufacturer. Another important concern is that in case of using captured data for tracking or for identifying people motion patterns, a history record of unique devices that traverse several APs will be required. Therefore, APs will need to be synchronized, which is not a straightforward task, and will have to cooperate, thus increasing complexity. Finally, and most importantly, since iOS 8, Android 6, and Windows 10, the 48-bit MAC addresses in mobile handsets are usually randomized. MAC randomization does not follow a universal policy and the specific randomization procedure is particular of the operating system, the driver, the WiFi chipset, and the hardware features of the mobile handset. Consequently, assuming that the same mobile device will have a global unique MAC address in all capture data might not be correct. Once a WiFi device is connected to a network, it uses the same MAC address (please note that it could be the physical real MAC or a non-real one obtained from a predefined calculation). However, as long as the device is not connected to an AP, the MAC (real or not) could vary. Therefore, some additional technique will be required to overcome this situation. Some interesting works addressing this issue can be found in [7] [8].

On the other hand, the rapid urbanization process that we are contemplating has meant an improvement and modernization of

This work has been supported by the Dirección General de Tráfico, Ministerio del Interior (Spain), project grant SPIP2017-02230 (STREET).

our lives, but also brings with it important challenges: traffic congestion, increased energy consumption, noise pollution, air pollution, and even the dehumanization of urban spaces. Nowadays, finding solutions to these problems is more viable thanks to new intelligent traffic management strategies that employ Information and Communication Technologies. Urban Computing [9], defined as the process of acquisition, integration, and analysis of heterogeneous data from different sources in an urban environment appears in this context to address the aforementioned great challenges to which cities must confront. In previous works [10] [11], we presented an original method for traffic control optimization in signaled intersections with notable performance outcomes. Our system was able to obtain the best time intervals for the traffic lights controlling an intersection using as input parameter the queue lengths (in vehicle units), with the particularity of detecting incipient traffic congestion. In this paper, we present a passive method to estimate the number of pedestrians in a signaled traffic intersection using WiFi technology. The goal is to get an estimation of how many pedestrians are waiting to cross in any of the pedestrian crossings of an urban arterial. The obtained value can be then used as additional input data into our traffic control optimization algorithm, together with the current traffic situation. By doing so, our system could adapt better to the road scenario taking into account more actors (i.e., vehicles and pedestrians). Given the paper length limitations, this work only focuses on introducing the pedestrian counting tool that we have developed.

The rest of the paper is organized as follows. Section II is devoted to describe our proposal. The methodology for the performance evaluation study of our proposal is included in Section III. Section IV shows the performance evaluation results. Related works are reviewed in Section V. The paper ends with a summary of the most relevant findings of this work in Section VI.

II. Description of the Proposed Method

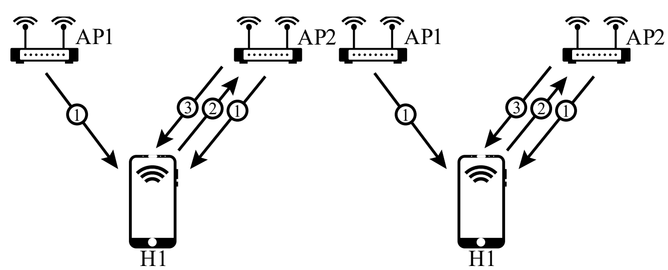
The IEEE 802.11 standard defines two methods for network searching, which are shown in Fig. 1. The first method is called Passive Scanning (Fig. 1 a). In this mode, the mobile device is in a listening state waiting for Beacon Frame messages. APs are in charge of sending these messages periodically (typically 100ms) to announce their presence. After receiving this message, the WiFi mobile device requests an association to the AP. This method is rarely used because the connection process has a high latency and several vulnerabilities have been found that could affect the mobile. The second method is called Active Scanning (Fig. 1 b). Contrary to the previous mode, the mobile device asks for WiFi networks. It sends Probe Request messages and waits for Probe Response messages from the AP. To do this, the mobile device sends the Probe Request messages to all WiFi channels successively, waiting for the response of the AP a maximum time called Maximum Channel Time. This mode allows saving energy and offers a faster connection to the AP. Active Scanning can be actually done in two different ways, namely, Direct Probe Request and Null Probe Request. In the former, the mobile device searches for a specific SSID (not used in practice due to several vulnerabilities [12], see e.g., Wardriving, WiGLE -Wireless Geographic Logging Engine-). In the latter, the mobile device asks all the SSIDs within a broadcast channel using a null SSID (ff:ff:ff:ff:ff), also called Wildcard SSID.

Because the Active Scanning process is the main mechanism used by mobile devices to search for WiFi APs, we will use it to estimate the number of pedestrians in signaled traffic intersections. We assume that each pedestrian carries a smartphone that includes a WiFi module and complies with the IEEE 802.11 standard. Probe Request messages contain information about the device that sends it and are sent unencrypted. Therefore, Probe Requests can be captured and analyzed. The format of a Probe Request message is shown in Fig. 2. Although the WiFi scanning process is described in the standard, there is not a standardized procedure that regulates how Probe Request messages are sent. Indeed, it depends on each cell phone [13], its current state (e.g., screen on/off, time in use, etc.), its operating system, manufacturer, etc.

A. Initial observations

As we have already said, our goal is to estimate the number of pedestrians waiting to cross a time-signaled intersection in order to adapt the time intervals of the traffic lights and minimize traffic congestion. To do this, a new input parameter corresponding to the people waiting to cross the intersection will be added to the previously proposed optimization algorithm [10]. The pedestrian counting algorithm should be able: 1) to differentiate static pedestrians waiting to cross versus pedestrians walking, 2) to differentiate pedestrians waking towards a specific intersection i versus pedestrians moving away from intersection i, and 3) to locate pedestrians waiting to cross into the appropriate pedestrian crossing. We assume that pedestrians walk at 1.39 m/s.

Before designing the pedestrian counting algorithm we carried out several simulations to get some knowledge about the characteristics of the Probe Request messages sent by pedestrians' WiFi mobile devices. Specifically, we employed computer simulations using OMNeT++ [14] to represent the behavior of



(a) Passive Scanning: (1) Beacon Frame; (2) Assoc. Request; (3) Assoc. Response

(b) Active Scanning: (1) Probe Request; (2) Probe Response; (3) Assoc. Request; (4) Assoc. Response

Fig. 1. WiFi scanning in 802.11. a) Passive Scanning and b) Active Scanning.

MAC header						Frame body		
bytes: 2	*2*	*6*	*6*	*6*	*2*	*Variable*	*Variable*	*4*
Frame Control	Duration	DA	SA	BSSID	Seq-ctl	SSID	Supported Rates	FCS

Fig. 2. Probe Request frame. DA ≡ Destination Address; SA ≡ Source Address (MAC); BSSID ≡ SSID of destination Access Point; Other client's pararameters such as supported rates, extended rates, sequence number, fragmentation number, etc.

64 pedestrians. The simulated scenario is shown in Fig. 3. For simplicity, we used a signaled intersection with four arms, each regulated by a traffic light. On top of each traffic light there is a Data Acquisition Unit called DAU_{ij}, where i represents the intersection and j represents the traffic light in intersection i. Each DAU can act as an AP, capturing Probe Request messages. The initial motion pattern of all pedestrians was as follows. In the time interval t given by 0 s $\leq t <$14.5 s (corresponding to timestamp *T0*), pedestrians walked from the left boundary of the simulated scenario marked as 1 in Fig. 3 to the pedestrian crossing marked as 2 in Fig. 3. Then, they waited 30 seconds (from 14.5 s $\leq t <$44.5 s) for the traffic light to turn green (timestamp *T1*). Afterwards, they crossed the intersection all the way to the right side of the simulated scenario (marked as 3 in Fig.3) from 44.5 s $\leq t <$74.5 s (timestamp *T2*). Fig. 4 represents the RSSI as obtained from the capture messages at DAU_{ij} (i=1, j={1, 2, 3, 4}) for pedestrian #1 along the observed periods of time. It can be seen from this figure that in the second time interval (14.5 s $\leq t <$44.5 s), when the pedestrian was static waiting to cross, the power) was in average more steady and higher in the DAU closer to the pedestrian, i.e., DAU_{11}. In the third time interval (44.5 s $\leq t <$74.5 s), the power presented a temporal variation in all DAUs, being more pronounced in the DAU that pedestrian #1 traversed, i.e., DAU_{12}.

For a better understanding, Fig. 5 depicts all RSSI obtained at each DAU_{ij} for the 64 pedestrians. If pedestrians were static, waiting to cross, the received power had few variations in all DAUs that captured Probe Request frames from the pedestrians, as shown in the time interval 14.5 s to 44.5 s in Fig. 5 for all DAUs. Indeed, the variation was smaller than 30 dB during that entire interval. As expected, if pedestrians were walking towards intersection i, the average power also increased along the observed time interval; see for instance Fig. 5 a) from t=0 s to

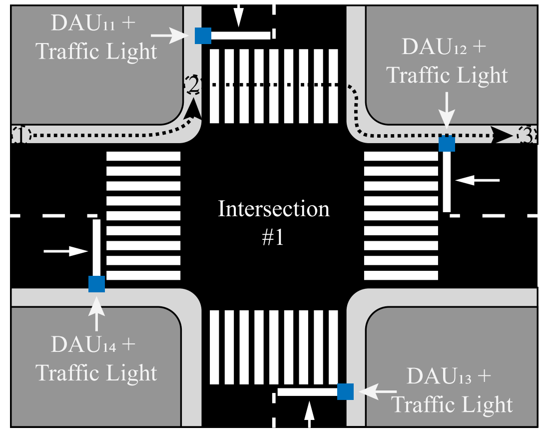


Fig. 3. Mobility pattern followed by the 64 pedestrians in an intersection.

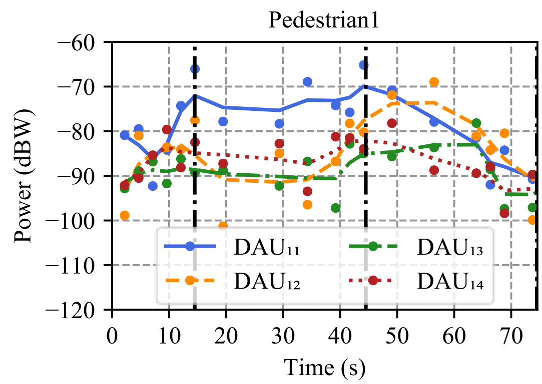
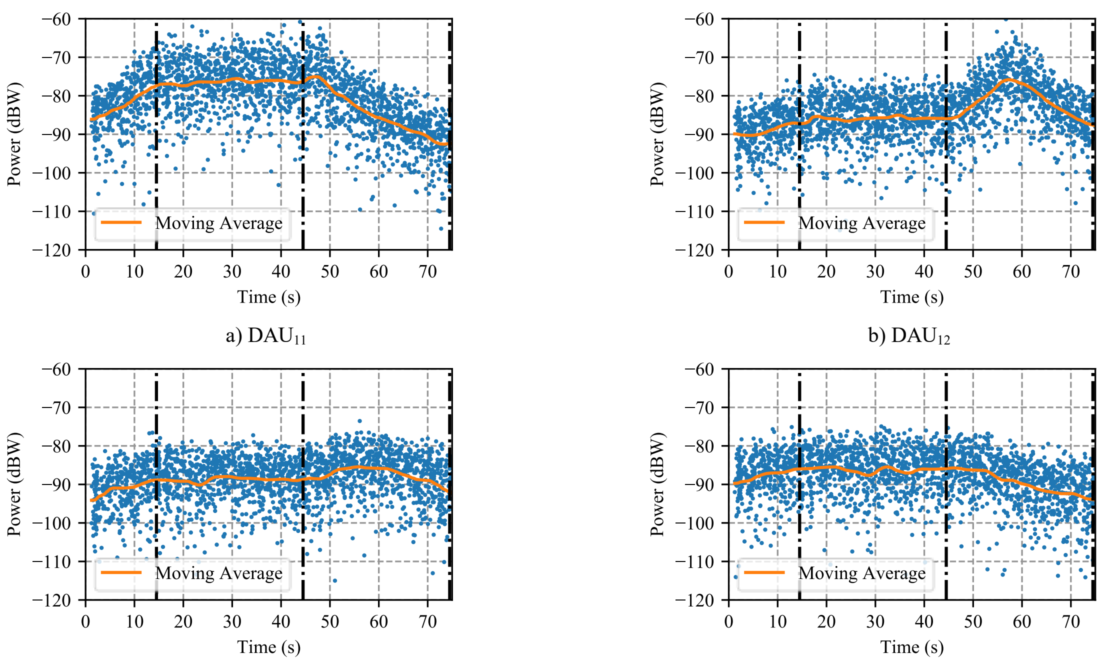


Fig. 4. Representation of the RSSI and the moving average for one pedestrian in diferent time intervals observed while walking through the scenario shown in Fig. 3.



a) DAU_{11}

b) DAU_{12}

c) DAU_{13}

d) DAU_{14}

Fig. 5. RSSIs and moving averages obtained from the capture Probe Request messages of 64 pedestrians carrying a smartphone with an enabled WiFi module in the simulated scenario represented in Fig. 3

t=14.5 s. The increase was more noticeable in the DAU closer to the pedestrians (DAU_{11}). On the contrary, if pedestrians moved away from intersection *i* the power decreased; with the exception of the DAU where pedestrians crossed the pedestrians crossing (DAU_{12}), which showed an increase and then a decrease in the measured RSSI (please see Fig. 5 b from *t*=44.5 s to *t*=74.5 s). Finally, the exact location where pedestrians were waiting to cross at intersection *i* can be obtained by observing the DAU with the highest average power throughout the capture interval.

B. Our proposal

From the initial observations, the general procedure of our algorithm is as follows. Each mobile device is treated individually, i.e., its behavior is studied through the total data-capturing time interval. This time interval corresponds to the time that a traffic light is in red phase for pedestrians. After this red phase, data captured by all DAU_{ij} are sent to one DAU. For example, all DAU_{1j} j={2, 3, 4} send capture data to DAU_{11} at intersection *i*=1. Then, DAU_{11} is responsible for calculating from all received data the number of pedestrians at each pedestrian crossing at intersection *1* and for introducing these values into the optimization algorithm that determines the duration of the next phase of traffic lights in this intersection. As we saw in Fig. 4, Probe Request messages from a particular mobile device *m* can be captured by several DAUs. After collecting all Probe Request messages in DAU_{11}, the pedestrian carrying the mobile device *m* is identified as *moving* if the power measurements of mobile *m* show a large variation in any of the DAU_{1j}. In case the variation in power measurements is small along the time interval, then the mobile device *m* (and so the pedestrian) is assumed to be *static* and *waiting* to cross under the DAU_{1j} that presents the highest power on average throughout the time interval.

From empirical tests, we observed a better performance of the algorithm if the random components of the wireless communication channel were mitigated instead of treating the power measurements (RSSI) as raw data. To do so, we decided to smooth the power received from the Probe Request messages with a Gaussian filter with a variable standard deviation. To get the optimal gaussian bell size, a processing of the results obtained via computer simulation was performed looking for the size that maximized the accuracy of the algorithm in full (*std_len*). That is, if the bell was very narrow, it did not show any filtering of the input data, the data stayed the same and showed a high variation. If the bell was very wide, the data was strongly smoothed and a straight-line equivalent to the average of the measured power was obtained. However, with an optimal value of the bell width, smoothing mitigated the variations in power measurements for situations where the pedestrian was static, while maintaining variations in power measurements in situations where the pedestrian was moving.

In sum, after grouping all captured Probe Request messages from one intersection *i* at the predefined DAU_{ij}:

- Pedestrians are differentiated based on the MAC address included in the Probe Request messages.
- For each pedestrian *p*, a list of DAUs DAU_{ijp} that have detected that pedestrian *p* is created and for each DAU_{ijp} the smoothed power measurements are stored.
- If any DAU_{ijp} observes a variation of power measurements higher than *var_max*, it is assumed that pedestrian *p* is moving. Therefore, it is not taken into account for the estimation of pedestrians waiting to cross. Otherwise, if the power measurements have a variation smaller than *var_max*, we assume that pedestrian *p* is static.
- If pedestrian *p* is static and detected as so in more than one DAU, then his correct location corresponds to the DAU_{ijp} that presents the highest power in average.

The pseudocode of the proposed method is depicted in Fig. 6.

Our proposal

```
The list_of_pedestrians detected by all DAUij is obtained.
for each pedestrian in list_of_pedestrians:
    the list_of_DAUij that have detected the pedestrian is obtained.
    static = True # by default the pedestrian is considered as static.
    for each DAUij in list_of_DAUij
        the power measurements of the pedestrian captured by the
        DAUij are obtained. This power measurements are
        smoothed.
        if var(smoothed) > var_max:
            static = False # now the pedestrian is considered as
              moving
    if static == True #this pedestrian is considered static in all
    list_of_DAUij then his correct location corresponds to the
    DAUij that presents a higher power in average.
    DAUij_max = getDAUijpMaxMeanPower(pedestrian)
    DAUij_max.addPedestrian(pedestrian)
```

Fig. 6. Pseudocode of the proposed algorithm.

III. Methodology for the Performance evaluation

This section describes the scenarios employed for the performance evaluation of the pedestrian counting tool introduced in this paper. The selected simulation tool was Omnet ++ [15]. The scenario for computer simulations is shown in Fig. 7. It consisted of an urban arterial with two intersections *i*={1, 2}. Pedestrians can be waiting to cross, walking towards a pedestrian crossing, crossing the intersection, moving to the next intersection, etc. The sidewalk was six meters wide, the width of the road was 15 meters, and the height of the buildings was 30 meters. Mobile nodes, which corresponded to *AdhocHost* in Omnet++, were WiFi devices with the Active Scanning on, sending Probe Requests periodically, and designed to simulate the different pedestrians' behaviors. We assumed that each mobile device represents a single pedestrian. On the other hand, DAUs were APs that capture Probe Request messages sent by the mobile nodes and were placed on top of the traffic lights, being in turn located at the vertices of the pedestrian zone that borders each intersection. In Omnet++, DAUs were represented as *WirelessAPWithSink*, modeling 802.11 WiFi Access Points with a sink. There were four DAU_{ij} i = {1, 2} j = {1, 2, 3, 4} at each intersection. Other characteristics used for the simulation study are included in Table I.

Four different sets of simulation tests were carried out for the performance evaluation study. First, the number of pedestrians in the simulated scenario was exponentially increased to assess

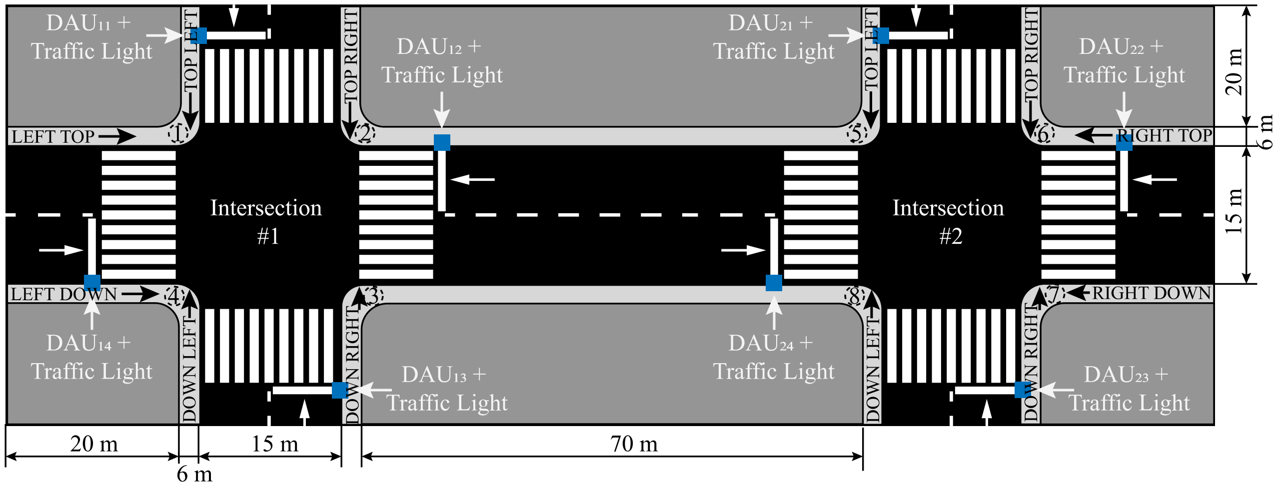


Fig. 7 .Scenario for the simulation analysis.

TABLE I. SIMULATION PARAMETERS USED IN OMNET++

Parameters	**Value**
Simulation tool / Framework	OMNetT++ / INET
Version	5.2.1 / 4.0
Ground type	Flatground
Obstacle loss	Dielectric Obstacle Loss
Propagation loss	Rayleigh Fading
DAU location in height	6 m
Mobile devices location in height	1.5 m
Mobile devices motion speed	1.39 m/s
Number of mobile nodes	64
Probe Request Period	2 secs
Transmission power	13 dBm
Reception sensibility	-120 dBm
Probability distribution to send Probe Request frames	Normal distribution with variable mean and variances

the influence (if any) of the number of pedestrians in the proposal. Then, the optimal values *var_max* and *std_len* for the smoothing process were calculated. Afterwards, the accuracy of the algorithm in counting and locating pedestrians was assessed allowing pedestrians to move in different directions and combinations. Finally, the impact of the frequency rate of sending Probe Request messages on the accuracy of the system was also studied.

IV. RESULTS

The first set of simulations showed that the number of pedestrians did not influence the radio channel. In other words, power measurements done by the DAUs (RSSI of the captured Probe Request messages) were not impacted by the number of mobile devices in the evaluated scenario. As an example, Fig. 8

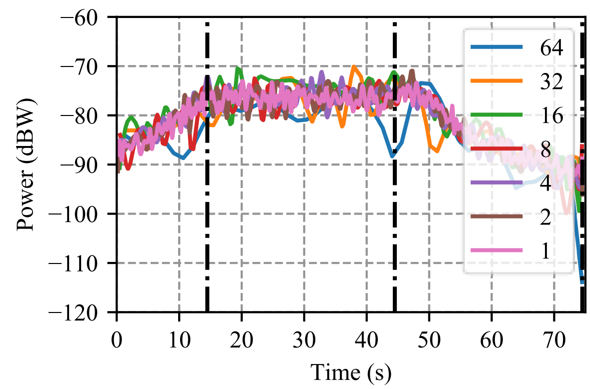


Fig. 8. Received power in Probe Request frames captured by DAU_{11} with an incremental number of pedestrians from 1 to 64.

represents the measured power of the capture frames under different conditions, with the number of pedestrians (mobile devices) increasing from 2^0 to 2^6. It can be seen that, in all cases, the power received by the DAU followed the same behavior disregarding the amount of people.

To apply the algorithm is necessary to define the optimal values for the *var_max* and *std_len* parameters. These are used in the smoothing process to mitigate the random components of the radio channel, and thus to differentiate easily static pedestrians from pedestrians in motion. A set of 64 pedestrians were simulated moving from left l to right r, traversing both intersections, using the top t sidewalk in Fig. 7 (this direction will be called D_{lrt}). After a grid search, the obtained results are included in Fig. 9. The selected values for *var_max* and *std_len* are 4.20 dB and 0.78, respectively.

In the third set of simulations, groups of pedestrians moved in several possible directions, specifically twelve:

1) from left l to right r in the top sidewalk t (crossing the two intersections), namely direction D_{lrt}, and from left l to right r in the down sidewalk d (crossing the two intersections), namely direction D_{lrd}

2) from right to left in the top sidewalk (crossing the two intersections), namely direction D_{rlt}, and from right to left in the down sidewalk (crossing the two intersections), namely direction D_{rld}

3) from top to down traversing only intersection i=1 in the left sidewalk, namely direction D_{tdl1}; idem in the right sidewalk, namely direction D_{tdr1}

4) from down to top traversing only intersection i=1 in the left sidewalk, namely direction D_{dtl1}; idem in the right sidewalk, namely direction D_{dtr1}

5) from top to down traversing only intersection i=2 in the left sidewalk, namely direction D_{tdl2}; idem in the right sidewalk, namely direction D_{tdr2}

6) from down to top traversing only intersection i=2 in the left sidewalk, namely direction D_{dtl2}; idem in the right sidewalk, namely direction D_{dtr2}

Each pedestrian group was composed by 64 pedestrians (mobile devices), representing a total of 768 pedestrians in the simulated scenario. For simplicity, all pedestrians were either static or in movement as shown in Table II and Table III. Under this complex scenario, the goodness of our proposal was verified. We defined three types of accuracies: A_{moving} represents the accuracy achieve by the whole system in detecting pedestrians who are moving, A_{static} represents the accuracy achieved by the whole system in detecting pedestrians in a static state (waiting to cross), and $A_{positioning}$ represents the accuracy achieved by the whole system in locating pedestrians waiting to cross into the proper pedestrian crossing (i.e., to know their exact location, the DAU_{ij} where they are waiting). These accuracies are calculated as indicated in (1), (2), and (3), respectively. The obtained values are depicted in Fig. 10. We can observe in this figure that despite the simplicity of the algorithm, we successfully identified pedestrians in motion with an accuracy above 52%. Static pedestrians were identified with an exactitude above 61%, and those pedestrians detected as waiting were recognized in the proper location with an accuracy above 93%.

$$A_{moving} = \frac{pedestrians\_detected\_as\_moving}{\mathrm{total\_number\_of\_pedestrians\_moving}} \quad (1)$$

$$A_{static} = \frac{pedestrians\_detected\_as\_waiting}{\mathrm{total\_number\_of\_pedestrians\_waiting}} \quad (2)$$

$$A_{positioning} = \frac{pedestrians\_located\_properly}{pedestrians\_detected\_as\_waiting} \quad (3)$$

TABLE II. TIMESTAMPS AND MOVEMENT STATES

Timestamp	Time Interval	State
T0	0s-14.5s	Moving
T1	14.5s-44.5s	Static
T2	44.5s-74.5s	Moving
T3	74.5s-104.5s	Moving
T4	104.5s-134.5s	Moving
T5	134.5s-164.5s	Static

TABLE III. EXAMPLE OF MOVEMENTES DLRT AND DTDL1 (D≡DIRECTION, L≡LEFT, R≡RIGHT, T≡TOP, D≡DOWN, 1≡INTERSECTION 1, 2≡INTERSECTION 2)

Movement	Example	Temporary Behavior
		0s - 14.5s: Left to (1)
		14.5s-44.5s: Static in (1)
horizontal	D_{lrt}	44.5s-104.5s: (1) to (5)
		104.5s-134.5s: (5) to (6)
		134.5s-164.5s: Static in (6)
		0s - 14.5s: Top to (1)
		14.5s-44.5s: Static in (1)
vertical	D_{tdl1}	44.5s-74.5s: (1) to bottom
		74.5s-134.5s: Bottom to top to (1)
		134.5s-164.5s: Static in (1)

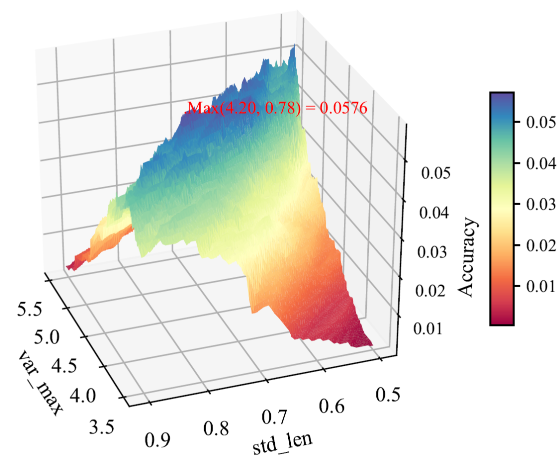


Fig. 9. Accuracy versus *var_max* and *std_len*. In this case, the acurracy shown is the product of *moving* and *static* accuracies at timestamps *T1*, *T2*, *T3*, *T4*, and *T5*, in order to maximize all acuracies.

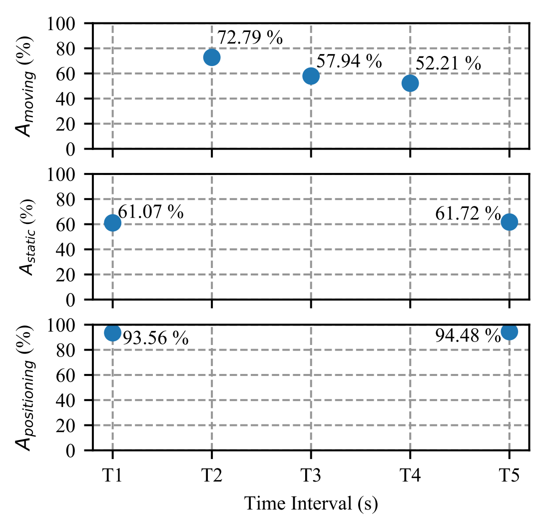


Fig. 10. Accuracies obtained in the simulated scenario shown in Fig. 7 with 768 pedestrians moving in twelve different directions. At timestamps *T2*, *T3*, and *T4* all pedestrians are moving, whereas at timestamps *T1* and *T5* all pedestrians are static.

A last set of simulations was performed to evaluate the impact of the Probe Request sending rate on the accuracy of our algorithm. Fig. 11 verifies that the lower the sending rate the lower the accuracy, since less frames are captured. However, the accuracy levels were kept above 51% for counting static pedestrians and above 95% for locating them if the time elapsed between two consecutive Probe Request frames is less than 16 s.

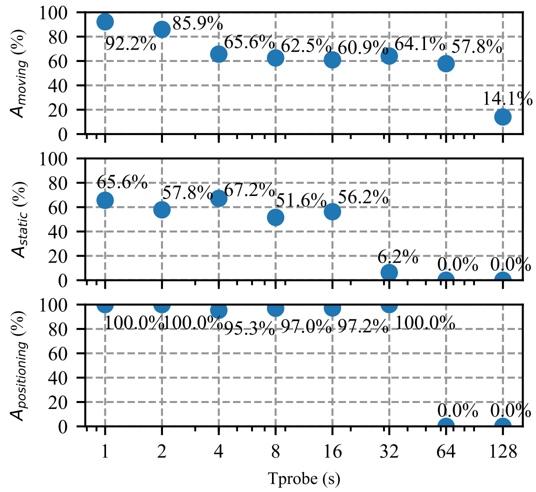


Fig. 11. Accuracies versus Probe Request sending rate (*Tprobe* represents the time interval between two consecutive Probe Request frames) in the scenario described in Fig. 3 for timestamps *T1* and *T2*.

V. Related Works and Discussion

Several WiFi-based passive methods for localization and tracking using Probe Request messages have been studied in the related literature. Some of the reasons to implement this type of systems are their initial simplicity and a low implementation cost, e.g., in development boards. In general, most works focus on indoor environments [1], [4]–[6], [15]–[18], achieving accuracy values around 75% in the best case; whereas for outdoor environments [19], [20], there are just a few works and they do not present results in terms of accuracy in counting pedestrians.

Redondi and Cesana conducted a survey in [1] about the use of Probe Request messages in two indoor scenarios for three different applications: location, user profiling, and device classification. The results obtained allowed locating a device with an error of 1.5m, as well as creating users' profiles and distinguishing between laptops and mobile devices. Also for indoor environments, Cianca *et al.* presented in [4] two WiFi based methods to count people. Their methods achieved an accuracy around 70% in the best case. In turn, the proposals presented in [5] and [6] employed WiFi and Bluetooth at the same time to estimate the number of people in indoor locations. In both cases, the accuracy obtained did not exceed 75%. In [15], the authors proposed to employ Probe Request messages to estimate the number of occupants in bus lines, obtaining an accuracy in the occupancy estimation around 60%. Similarly, Probe Request messages were used in [16]–[18] to locate people inside buildings, reaching the conclusion that the use of Probe Request messages was a viable solution to monitor smart buildings occupation and to extrapolate these data to other fields such as energy saving or surveillance.

If we focus on outdoor environments, there are just a few works in the related literature. An example can be found in [19], where the authors presented a real test bed for the space-time monitoring of pedestrians. The results obtained had a positioning error below 70 meters but no results in terms of accuracy in counting people were provided. This method could be useful for large deployments, but it is not applicable to optimize traffic signaled intersections. In a similar way, in [20] authors presented a system to analyze crowds' behaviors. Even though interesting outcomes were obtained regarding crowd mobility, no results in terms of accuracy in counting people were included.

As a consequence, we believe that our method is simple but comparatively robust enough. The achieved accuracy levels are comparable (and for some cases even better) than those obtained in indoor scenarios in the related literature. Above all, it is remarkable the exactitude that our method attains in estimating static pedestrians and locating them in the proper pedestrian crossing. It is important to note that our algorithm requires simplicity, so that it can be used in real time, and lightness, so that can be implemented in embedded devices. Specifically, the time interval that a traffic light is in a red phase (approximately between 30 s and 90 s) is the time that the algorithm has to be executed. Thus, we work with small sets of captured data. Another advantage of our proposal is the transparency from the pedestrians' perspective. In addition, our system does not require time coordination among the DAUs because is the Programmable Link Controller of the traffic light the one responsible for asking for the data every time there is a phase change (from red to green and vice versa). Nevertheless, we are aware of possible limitations and future improvements. As mentioned in Section I, there are several drawbacks of WiFi-based methods for people counting. For instance, MAC randomization should not be ignored. In the testbed where we are conducting real experiments we are evaluating different alternatives either to associate random MACs to the same device (not necessarily to the real MAC but to the same mobile device) or to quantify the relationship Probe Request frames with a random MAC versus real mobile devices.

VI. Conclusion

The role of pedestrians in urban environments is of vital importance to develop intelligent control traffic systems. Despite the abundance of people detection methods, WiFi-based approaches are gaining momentum due to their initial simplicity. However, practice has revealed important limitations to be overcome in order to achieve the expected levels of accuracy in outdoor environments. We have presented a WiFi-based passive method that is able to differentiate among static pedestrians (waiting to cross in pedestrian crossings) and moving pedestrians, and to provide the exact location of static pedestrians. The performance evaluation study has shown levels of accuracy higher than those achieved in the related literature. As future work, we plan to integrate the proposed counting tool with our previous optimization mechanism for traffic control in signaled urban intersections.

References

[1] A. E. C. Redondi and M. Cesana, "Building up knowledge through passive WiFi probes," *Comput. Commun.*, vol. 117, pp. 1–12, 2018.

[2] J. Garcia, A. Gardel, I. Bravo, J. L. Lázaro, M. Martínez, and D. Rodríguez, “Directional People Counter Based on Head Tracking,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 3991–4000, 2013.

[3] M. Stubenschrott, T. Matyus, and C. Kogler, “Real-Time Estimation of Pedestrian Inflow Rates from Saturated Sensor Counting Data in a Complex Metro Station,” in *Proc. IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 1958–1963.

[4] E. Cianca, M. De Sanctis, and S. Di Domenico, “Radios as Sensors,” *IEEE Internet Things J.*, vol. 4, no. 2, pp. 363–373, 2017.

[5] Y. Yuan, “Crowd Monitoring Using Mobile Phones,” in *Proc. 6th International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2014, pp. 261–264.

[6] L. Schauer, M. Werner, and P. Marcus, “Estimating Crowd Densities and Pedestrian Flows Using Wi-Fi and Bluetooth,” in *Proc. 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services MOBIQUITOUS’14*, 2014, pp. 171–177.

[7] M. Vanhoefy, C. Mattez, M. Cunchez, L. S. Cardosoz, and F. Piessensy, “Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms,” in *Proc. 11th ACM on Asia Conference on Computer and Communications Security ASIA CCS*, 2016, pp. 413–424.

[8] J. Martin *et al.*, “A Study of MAC Address Randomization in Mobile Devices and When it Fails,” *Proc. Priv. Enhancing Technol.*, vol. 4, pp. 268–286, 2017.

[9] Y. Zheng, W. Wu, Y. Chen, H. Qu, and L. M. Ni, “Visual Analytics in Urban Computing: An Overview,” *IEEE Trans. Big Data*, vol. 2, no. 3, pp. 276–296, 2016.

[10] R. Sanchez-Iborra and M.-D. Cano, “On the similarities between urban traffic management and communication networks: Application of the random early detection algorithm for self-regulating intersections,” *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 4, 2017.

[11] R. Sanchez-Iborra, J. F. Ingles-Romero, G. Domenech-Asensi, J. L. Moreno-Cegarra, and M.-D. Cano, “Proactive Intelligent System for Optimizing Traffic Signaling,” in *Proc. IEEE 14th International Conference on Pervasive Intelligence and Computing PICom’16*, 2016, pp. 1–6.

[12] N. Sidiropoulos, M. Mioduszewski, and E. Schaap, “Open Wifi SSID Broadcast vulnerability,” 2012. [Online]. Available: https://www.os3.nl/_media/2012-2013/courses/ssn/open_wifi_ssid_broadcast_vulnerability.pdf.

[13] J. Freudiger, “How talkative is your mobile device? An experimental study of Wi-Fi probe requests,” in *Proc. 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2015, p. 8:1-8:6.

[14] “OMNeT++ Discrete Event Simulator - Home Page.”. [Online]. Available: https://www.omnetpp.org/. [Accessed: 17-Jul-2018]

[15] W. Pattanusorn, I. Nilkhamhang, S. Kittipiyakul, K. Ekkachai, and A. Takahashi, “Passenger estimation system using Wi-Fi probe request,” in *Proc. 7th International Conference on Information Communication Technology for Embedded Systems IC-ICTES’16*, 2016, pp. 67–72.

[16] E. Vattapparamban, B. S. Çiftler, I. Güvenç, K. Akkaya, and A. Kadri, “Indoor occupancy tracking in smart buildings using passive sniffing of probe requests,” in *Proc. IEEE International Conference on Communications Workshops ICC16*, 2016, pp. 38–44.

[17] B. S. Çiftler, S. Dikmese, I. Güvenç, K. Akkaya, and A. Kadri, “Occupancy Counting With Burst and Intermittent Signals in Smart Buildings,” *IEEE Internet Things J.*, vol. 5, no. 2, pp. 724–735, 2018.

[18] L. Sun, S. Chen, Z. Zheng, and L. Xu, “Mobile Device Passive Localization based on IEEE 802.11 Probe Request Frames,” *Mob. Inf. Syst.*, vol. 2017 Artic, pp. 1–10, 2017.

[19] A. B. M. Musa and J. Eriksson, “Tracking unmodified smartphones using wi-fi monitors,” *Proc. 10th ACM Conf. Embed. Netw. Sens. Syst. - SenSys ’12*, p. 281, 2012.

[20] A. Basalamah, “Crowd Mobility Analysis using WiFi Sniffers,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 12, pp. 374–378, 2016.