Detecting Relative Position of User Devices and Mobile Access Points

Leonid Kholkine*, Pedro M. Santos*†, André Cardote‡, Ana Aguiar*†
*Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia, Universidade do Porto
†Instituto de Telecomunicações
‡VENIAM
Email: {ee05254, pedro.salgueiro}@fe.up.pt, acardote@veniam.pt, anaa@fe.up.pt

***Abstract*—Free Internet access in urban public transportation fleets by means of WiFi hotspots is becoming a common service in a growing number of cities. Users of the public bus WiFi service will get the network memorised in their mobile devices and, as they move from buses to streets, their devices will switch Internet access from WiFi to cellular. If a bus passes by the user while he or she is on the street, the mobile device will connect to the bus WiFi hotspot, breaking the cellular connection and initiating a short-lived WiFi connection. This disruption of the user experience can be avoided if the mobile access point (AP) learns whether the user device is inside or outside the bus and decides to accept its connection request or not. In this paper, we evaluate if a mobile AP (installed on the buses) can detect the relative position of a user device. We collected a large dataset of WiFi-usage and mobility patterns from a real-world public bus fleet in Porto, Portugal, to identify features from which the mobile AP can detect if a user device is inside or outside the bus. We tested RSSI and bus speed as features feed into a classifier, and observed that RSSI is a poor indicator whereas bus speed presents some correlation to whether a connection is established inside or outside the bus. Finally, we propose and evaluate the trade-offs of a mechanism (running on the mobile AP) that decides whether a device should be allowed to connect or not.**

***Index Terms*—Vehicular Access Point, User Experience, Positioning**

I. Introduction

Currently, the public bus fleet in Porto, Portugal, offers free WiFi to its passengers via access points (AP) using the same Service Set Identifier (SSID). It has been observed that users of mobile devices leave Internet access on (both WiFi- and cellular-based) when they are on the street. In the case a user has already used the WiFi access point of a public bus, the network will be memorised by the user device and connected to every time the device discovers it, even if sometimes the user is not on the bus. In such cases, the cellular connection will be broken and a new WiFi connection will be established to the mobile AP in the bus which lasts for a very short time. This latter connection is bound to cause a bad experience for the user, making it an undesired connection.

The undesired connection can be avoided if the mobile AP can learn the relative position of the user device and make a decision on whether to accept its connection request or not. In this paper we study if, while a user device tries to connects to the mobile AP, it is possible for the AP to decide whether a user is inside or outside the bus and allow the device to connect or not. The final goal is to develop a software implementation of a decision algorithm to be deployed at the mobile AP to mitigate the occurrence of undesired connections. The motivation to develop a solution for the AP, as opposed to targetting the user device, is a practical one: it is simpler for the mobile WiFi service operator to deploy a solution in the APs than to all users install a dedicated application.

Our contributions are:

- Characterization of such undesired connections and the conditions in which they are likely to happen;
- Development of a classifier for predicting the relative position of the mobile AP and a client device using real-world datasets;
- Design and implementation of a gatekeeper mechanism (decides whether client device should connect) in widely-used open network stack implementations.

The remainder of this article is as follows. In Section II, we outline the existing state of the art. A characterization of undesired connections is discussed in Section III. The development of a classifier to detect the relative position of the user is explained in Section IV. The design of a mechanism to identify relative position and decide on connection acceptance or refusal is presented in Section V. Final remarks and future work are laid out in Section VI.

II. Related Work and Background

The main option we found for an off-the-shelf mobile AP to localize a client device is to use RSSI-based solutions. RSSI-based methods for localization include lateration methods, machine learning classification, probabilistic approaches and statistical supervised learning techniques [1]. These localization algorithms assume locations suffer less change and allow multiple fixed APs to improve the localization estimate. The studies of [2], [3] show that, even under optimal circumstances, it is difficult to use RSSI to reliably measure a distance as results are not consistent among different conducted experiments. The work of [4] tries to infer user motion and location from WiFi received signal strengths. Concerning vehicular networks, a survey on the topic can be found in [5]. The work of [6], [7] was one of the first to propose a collaborative RSSI-based localization solution. Localization with infrastructural nodes has also been proposed: in [8], the authors use RSSI and angle-of-arrival from infrastructural APs to improve their

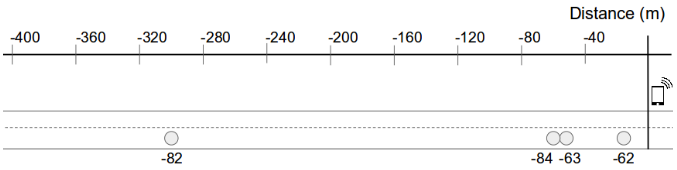

Fig. 1. RSSI and the position during the connection to the moving AP

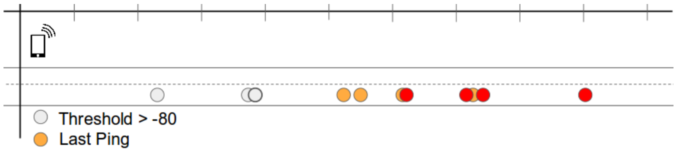

Fig. 2. Distance from the client when connection is lost

vehicle's position estimate. However, we have not found works addressing user localization by mobile APs. In [9], the RSSI of V2V messages is used to predict vehicle collisions. Some works focus on vehicle-to-pedestrian (V2P) communication for safety purposes. The authors of [10] propose WifiHonk, a beacon-based safety mechanism to avoid collisions. In [11], the authors describe an implementation of a DSRC stack in a smartphone-grade WiFi chip to enable V2P communication. The solutions imply both involved terminals (in our case, AP and client) to have dedicated software and/or hardware, which we want to avoid given that the solution should be limited to the AP (see motivation in previous section).

III. Characterization of Undesired Connections

Our first step was to characterize the undesired connections that we focus on – the case of a static WiFi client device on the road side connecting to mobile access point –, and how to distinguish them from "normal" connections (client is inside the bus). We conducted an experiment with a stationary user device (smartphone) on the road side and a moving access point to understand undesired connections. A second experiment sought to characterize the experienced RSSI when the client device is inside the bus. We describe next the methodology and results of both experiments.

A. *Characterization Experiments – Road-side Client*

We describe the experiments conducted to characterize undesired connections between a static road side client and the mobile access point. We want to characterize such connections at connection time – the instant that is relevant to us, in order to decide whether to accept or reject a connection – and across its entire duration. In particular, we sought to answer the following questions:

- At which distance does the mobile access point connect? What is the RSSI when the client device connects?
- How does the RSSI change when the client device is connected to a moving AP?
- At which distance does the RSSI disconnect from the AP?
- How long is the client device connected to the AP when it is moving?

The used methodology was as follows. A car has been equipped with an access point similar to those used in the public buses. The access point has a GPS module installed, which allows to learn the vehicle speed at the time of the connection attempt. It also allows to measure the RSSI at the time of the connection.

A client device – smartphone running Android OS – was positioned statically by the road side. In order to ensure that data is always exchanged and the RSSI is updated through out the time, a continuous ping was executed every second. The following information was recorded:

- Timestamp of established connection and connection loss on the client device;
- Timestamp of the first and last transmitted ping;
- Sets of RSSI samples and GPS coordinates taken simultaneously at a rate of 1 second, defined by the mobile AP's refresh rate of GPS information.

The experiment was done at night and in a straight street to ensure similar conditions across measurement sessions. Four tests were conducted at an average speed of 30 km/h on a best-effort basis (traffic lights and other traffic made it hard to maintain constant speed all the time). During the test the device was using an old IP address and only sending a DHCP Packet, therefore decreasing connection establishment time.

The results were as follows:

1) Distance at which connection occurs: The distances at which the mobile AP and the client device were when connection occurred and the RSSI values at connection time are represented on Figure 1. Looking at the distances between AP and client, the connection was established as close as 20 meters and as far as 300 meters away. This variation of the distance at which connection occurs is caused by the fact the Android device only performs a scan every 15 seconds by default. The RSSI at connection time presents little relation to the distance at which it occurs.

2) Distance at which connection is lost: A closer look at when the client device disconnected from the mobile AP is presented on Figure 2. It is shown that the client took some time before concluding that the mobile AP is not in range anymore. The loss of connection happened when the AP was from 240m to almost 300m away from the client. This shows that, if a stationary client device connects to a moving AP, there is a time period during which the device is mistakenly trying to communicate with the AP.

One important behaviour observed on the client device is that the connection does not switch from cellular to WiFi until the device acquires an IP address from the AP. From the client perspective, this means the WiFi connection is not effective until the DHCP process is complete.

3) RSSI values and duration of connections: Figure 3 shows the RSSI registered at the mobile AP, at different distances between AP and client device. It can be seen that even under similar conditions the RSSI values varied more than 20 dBm

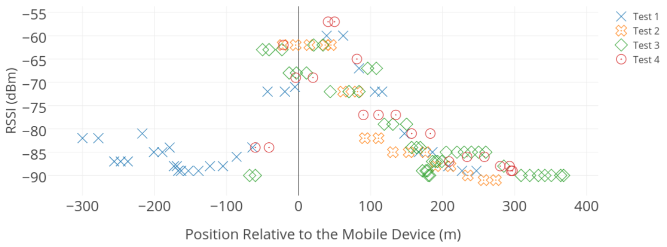

Fig. 3. RSSI at different distances during the characterization experiment

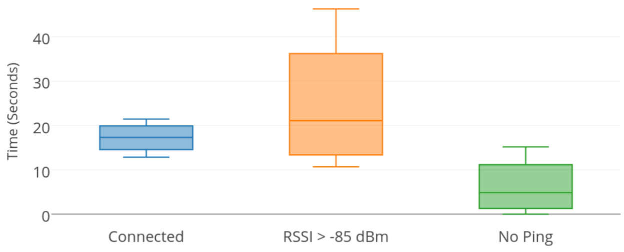

Fig. 4. Time spent connected to the AP

when the AP was close to the client. When farther away from the client, the RSSI values were more uniform.

The connection duration during the tests lasted from 32 seconds to 64 seconds. The duration of a connection can be separated into three classes of periods:

- *Connected* - When the RSSI value is above -85dBm (less than 75 meters away);
- *Poor connection* - When the RSSI value is below -85dBm (more than 75 meters away);
- *No Ping* - Period during which AP receives no pings.

The boxplots for time spent in each class are in Figure 4.

We can conclude from the initial tests that the stationary client spends a considerable amount of time with low RSSI value compared to when close to the AP. Regarding the total connection distance, during the tests the connection period never corresponded to more than 700 meters that the car has travelled.

B. *Characterization Experiments – Passenger Client*

To characterize regular connections and understand how they differ from undesired connections, we took RSSI measurements inside buses. Due to on-going WiFi service operation, the AP could not be used to record the clients' RSSI and connection duration. Instead, measurements were performed with a customized application for Android devices. The measurements were recorded in a normal and an articulated bus, at different locations inside the buses. We performed 10 different tests in each type of bus and with a different number of passengers. Each test was done for 30 seconds while sending a ping to the AP.

The results are shown in Figure 5. We can conclude that the RSSI values measured by the client device inside the bus stay within a certain range. Based on this, we hypothesize that a criterion for not allowing connections could be based on the RSSI of packets exchanged versus the RSSI range of connections inside the bus.

C. *Conclusions for Future Steps*

The second experiment provided the more insightful conclusion to tackle relative position identification, i.e., to have the mobile AP detect if a passenger is inside or outside the bus. We observed that RSSI values measured by a client device inside a bus do not stray considerably from a range of values, and we can use that information to distinguish users inside and outside the bus. This conclusion gives indication that RSSI is a strong candidate to distinguish both types of clients.

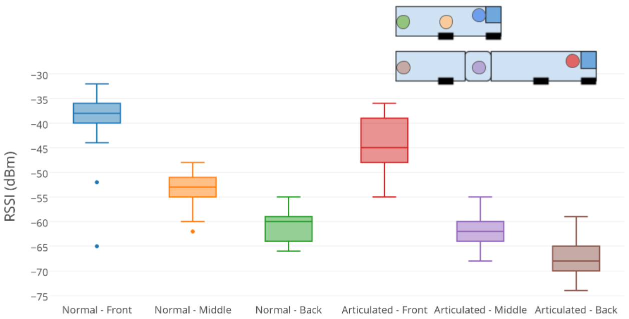

Fig. 5. RSSI measured inside buses.

IV. Developing the Classifier

We developed a clasifier to identify the relative location of the mobile AP and a client device. The features used by the classifier should be extracted from information available to the mobile AP, given that the classifier should be deployed in mobile APs. We started by collecting a large-scale dataset of bus GPS traces and RSSI samples of client devices collected at the mobile AP. An initial set of features was constructed based on the conclusions of the preliminary experiments (previous section), and we used the RapidMiner tool [12] for feature selection.

For the remainder of this section, we present the used dataset, the initial feature selection process, the decision tree that was created and associated parameters, a performance evaluation, and a discussion section on the resulting classifier.

A. *Real-World Connection Dataset*

We collected a large-scale dataset of connection and mobility traces from an urban bus fleet. The mobility traces and connection history between client devices and mobile APs was recorded during one week on seven city buses. The buses changed route randomly from one day to the next. The following information was stored for each packet received at the mobile AP: timestamp, MAC address, average RSSI value, GPS position, and GPS speed. Two packets with the same MAC address were considered to belong to the same connection when they appeared in the dataset with no more than 2 seconds difference between them. In case there was a difference bigger than 2 seconds, packets were considered to belong to a separate connection. In total, around 5 million lines of data were gathered, corresponding to 14063 connections. Data pre-processing consisted on discarding connections that: (i) had less than 10 samples; (ii) had failed GPS measurements during the connection. The resulting dataset contained 12040 connections.

The dataset did not have any ground truth annotation for our classification task – whether the connecting device is outside or inside the bus. So, it was necessary to find a criteria to differentiate between the two types of connection. Reasoning that a device connected outside the bus will not keep a connection across a long distance, we inferred the ground truth based on the distance travelled by the bus during a connection and its duration. We did this by searching for a threshold for the distance travelled by the bus that would produce the most separable categories in terms of total connection duration. We emphasize that none of these quantities is suitable as feature for the classification task because they refer to the whole connection, while the features must refer only to inital periods to enable a real-time decision. We considered four arbitrary thresholds for the distance travelled by the bus: 250, 500, 750 and 1000 meters. For each of the categories, outliers on the duration of the connection were removed using the Hampel identifier. The results for each group are presented in Figure 6. The 250 meters threshold produces the best separation between the two categories, because there is no overlap in connection duration. I.e., we have either brief and short connections (outside the bus) or long lived and long connections (inside the bus). Thus, we separated the dataset into two classes of connections:

- *Outside*: connections during which the bus travelled less than 250 meters, and connection lasted between 20 seconds and approximately 2 minutes;
- *Inside*: connections during which the bus travelled more than 250 meters, and connection lasted between approximately 4 and 7 minutes.

We ended up with 2350 connections outside the bus and 7400 connections inside the bus.

B. Feature Selection

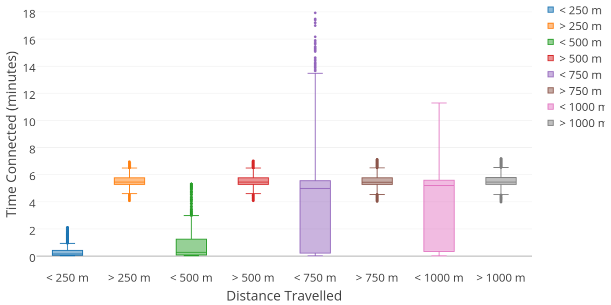

Fig. 6. Data segmentation, inside vs outside the bus, for distances with outliers removed using the Hample identifier

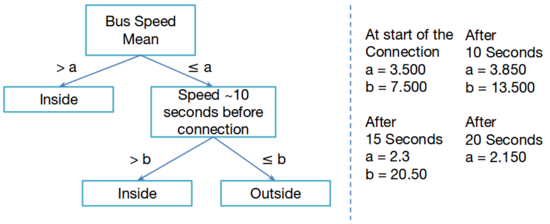

Fig. 7. Summary of the decision trees generated for different time intervals

The set of features we considered was divided into two classes, RSSI-based and speed-based, as these are the data that the AP has access to. We emphasise that we can only consider features that are computable in real-time from the initial packets of a potential connection, since we aim at deciding whether to let the connection proceed or not. We started by selecting the RSSI and speed at connection time. We then included the average and variance of speed and RSSI for N seconds after the start of the connection. Choosing a value for N implies a practical trade-off when the classifier is implemented in the mobile AP: the longer the time interval to decide whether the user is inside or outside the bus, the longer the user has to wait before being given Internet access. Finally, we thought that the bus speed bus speed prior to the connection start might be relevant, since it would express e.g. whether the bus had just been at a stop. Since we only collected the speed during the connection, this feature was extracted from other on-going connections on the same bus in the relevant time window.

Overall, the features we generated were the following:

- RSSI at connection time;
- RSSI average for the first N seconds of connection;
- RSSI variance for the first N seconds of connection;
- speed at connection time;
- speed average for the first N seconds of connection;
- speed variance for the first N seconds of connection;
- speed average of the N_{prev} seconds prior to connection;
- speed variance of the N_{prev} seconds prior to connection

The time windows N (in seconds) used to construct features were 5, 10, 15 and 20 seconds. The speed prior to the connection feature was calculated using an arbitrary time interval N_{prev} of 10 seconds.

C. Classifier Training and Performance Evaluation

We opted for a decision tree classifier due to the simplicity of the task and the need to later implement the solution in software on an embedded device. The obtained final dataset of 2350 connections outside the bus and 7400 connections inside was further sub-sampled to obtain 2350 connections of both classes. The connections were split into 70% for training the decision tree and 30% for testing. RapidMiner was run separately for each N in the set, {5s, 10s, 15s, 20s}.

The decision tree output by RapidMiner is shown in Figure 7, with the produced parameters values for the different N. Respective accuracy and true positive and negative rates are shown in Table I. Three main conclusions can be drawn.

1) RSSI was discarded: RSSI metrics were completely discarded by the feature selection process and only speed features were left. We discuss this in more detail in Section IV-D.

TABLE I
SENSITIVITY, SPECIFICITY AND ACCURACY OF THE DECISION TREES FOR DIFFERENT TIME INTERVALS

	Initial info.	After 10s	After 15s	After 20s
Inside the Bus (Sensitivity)	73.59%	80.77%	83.85%	87.44 %
Outside (Specificity)	63.85%	61.03%	56.92%	55.90 %
Accuracy	68.72%	70.90%	70.38%	71.67 %

TABLE II
SENSITIVITY, SPECIFICITY AND ACCURACY OF THE DECISION TREE WITH INITIAL INFORMATION AND AFTER 20 SECONDS WITH PREVIOUS SPEED

	Initial Info.		After 20s	
Prev. speed	without	with	without	with
Inside the Bus (Sensitivity)	65.38%	73.59%	81.22%	87.44 %
Outside (Specificity)	67.93%	63.85%	61.31%	55.90 %
Accuracy	66.79%	68.72%	71.41%	71.67 %

2) Larger time windows improve classifier accuracy: This is explainable by observing that, if a high bus speed is recorded over an increasingly larger connection duration, the more probable it is that the client is inside the bus. The contrary may not be true: if the connection is lasting substantial time but the bus is stopped or at low speeds, the user can be either inside or outside the bus, e.g. by the bus stop or a traffic light. One exception to this is when the user is actually outside the bus but at a similar speed (e.g. in a nearby car). However, these connections should be considered inside the bus according to our ground truth inferrence, and rightly so because no quality of experience impairment should be expected.

3) Prior speed is a relevant feature: Speed prior to connection was selected as a feature when the bus speed is low. We can conclude that knowing if the bus has just stopped influences the decision about whether the client device is inside or outside the bus. This feature provided considerable performance improvement. In Table II, we show the results with and without this feature to emphasize the performance increase it brings.

Finally, we explore the trade-off between true positives and true negatives and the time to connect. We combined the decision trees produced for the different N into a single procedure, shown in Figure 8, and ran it in RapidMiner. The results for the percentage of the users that are allowed to connect or not is shown on the Figure 9. We see that we could increase the percentage of the users that are inside the bus and are allowed to connect (true positive rate) by 10% at the cost of some connection delay (10 s) and a 4% increase in false positives.

D. Why RSSI is not a good feature

We looked in detail at the RSSI samples to understand why it was not a chosen feature. We sought the differences in RSSI behaviour among connections outside and inside the bus. As there are many data points, we restricted our study to the mean, median and standard deviation during the first 20 seconds of connection.

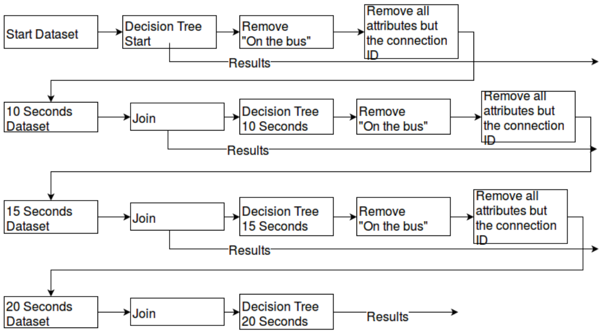

Fig. 8. Chained processing combining all decision trees.

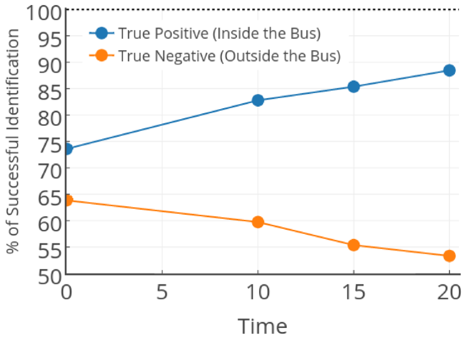

Fig. 9. True and False positives for different time intervals.

The RSSI mean and median can be visualized in Figure 10. It is possible to see that there is little to no difference between the RSSI inside and outside the bus. Figure 11 shows the mean and median of the variation of the RSSI, and again no difference can be seen for the data when a user is inside or outside the bus.

We hypothesize that the results presented above are a consequence of different devices producing different RSSI levels, as shown in [13]. Because the first half of the MAC address is assigned to the chipset manufacturer by the IEEE Standards Registration Authority (Assignment), it is possible to analyse the RSSI median by manufacturer. In order to see the influence of the manufacturer on the RSSI level, we grouped the connections by Assignment code and a boxplot of the median of the RSSI can be seen in Figure 12. Only the connections that travelled for over 1 kilometer of distance were included in the plot, and only if there was more than 40 connections for the same Assignment. The result supports our hypothesis that RSSI levels vary considerably among chips of different manufacturers, contributing to the irrelevance of RSSI as a feature in this classification task.

V. GATEKEEPER SOLUTION

We now describe a solution that implements the classifier in practice. The gatekeeper mechanism is to be installed in the mobile APs and evaluate if a mobile device is inside or outside the bus before it is allowed to connect, using the classifier described in the previous section. A preliminary study was conducted to identify the best occasion to enforce

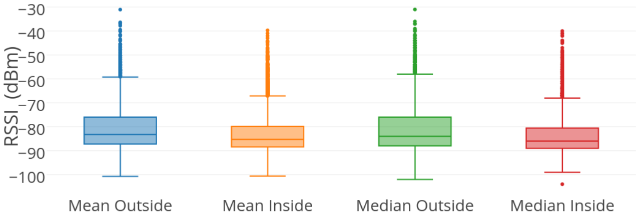

Fig. 10. The RSSI mean and median

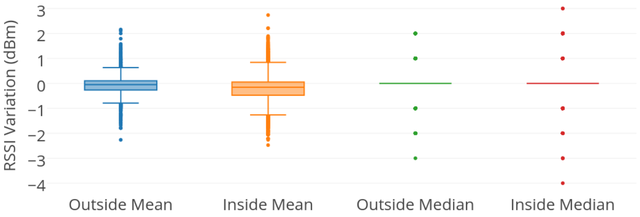

Fig. 11. The mean and median of the variation of the RSSI

the gatekeeper mechanism; it is presented in the following subsection. Details on the proposed implementation using the prior conclusions follow in the remaining subsections.

A. Preliminary Study: Impact of Connection Denial

After deciding whether an associating device is inside or outside a bus, it is necessary to implement a mechanism to avoid connections from devices in the second category. For that purpose, we search to understand the behaviour of a user device when it is denied a connection to an AP, to see when it is the best occasion to perform the denial. We start by briefly reviewing the stages required for a mobile device to connect to a WiFi access point: Discovery, Authentication, Association and IP assignment. During discovery stage, the mobile device searches for an AP actively or passively. Once an AP is found, the mobile device initiates two message exchanges leading to Authentication and Association to the AP. Regarding DHCP operation, the IP assignment process can take several seconds due to two reasons [14]: i) if the devices find WiFi networks with an SSID previously seen, the devices will try to renew the IP address lease; and ii) the DHCP server performs a Duplicate Address Detection (DAD) to ensure the offered address is not in use. As mentioned in Section III-A, the previous Internet connection is only broken when the mobile device has successfully acquired an IP address on the WiFi connection. Thus, the connection can be denied at the stages that precede IP assignment – WiFi authentication and association – or at the DHCP IP assignment process itself. In any of the phases, the AP can deny access either explicitly, by sending a message, or implicitly, by not responding. Unfortunately, the behaviour of a mobile device in either case for the three phases is not standardized.

With this information in mind, we designed a test to understand how a mobile device behaves when it is denied connection to an AP. The test was conducted on 3 smartphones with different Android versions: Lenovo Vibe Shot (Android 6.0.1), LG Nexus 4 (Android 4.4.4) and Samsung S3 (Android 4.3). The test was conducted with modified version of *hostapd* that either did not reply to requests or denied them, to emulate the available options. During one minute, counting from the first authentication packet received, the timing of the packets was recorded and the duration between packets was calculated.

The results can be seen in the Figures 13 to 17. We observed that mobile devices can behave very differently depending on the model of the mobile device used. For example, when denied an Association request (shown in Figure 14), the Samsung S3 sends a packet every 4 seconds, and the LG Nexus sends 3 packets with a 30 seconds interval between the second and third packets.

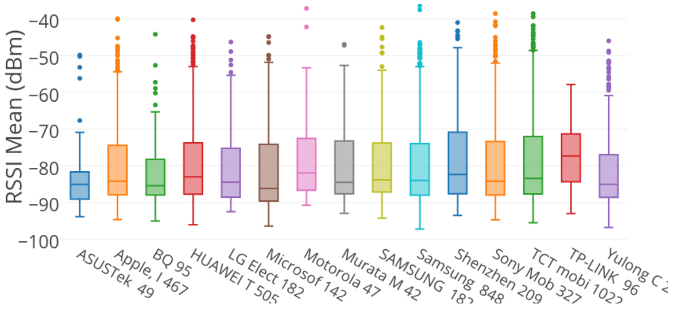

Fig. 12. RSSI Values per Manufacturer

From thus study, we conclude that the DHCP stage is the preferential stage to deny the connection. As seen in Figure 17, ignoring DHCP packets causes the fastest retransmission in all devices, even though it may grow to inter-packet periods of up to 16 seconds. Moreover, the behaviour is more similar among different mobile devices: the time is increased between packets until a certain threshold is reached, and afterwards it restarts from a lower value. In any case, the time between packets received when denying a connection is long and does not allow to sample different RSSI values, making its estimation more difficult.

B. Gatekeeper Overview

We concluded above that the DHCP stage is the best to deny the connection as the behaviour is the most consistent across different mobile devices. Therefore, we incorporated our mechanism in the Linux kernel DHCP server code *dnsmasq*. This allows us to have direct access to the DCHP probes from client devices that arrive from the network.

Our approach to develop the gatekeeper mechanism is that we wish to keep the delay experienced by the users inside the bus to a minimum, despite the conclusion of Section IV that a longer period of data collection results in better accuracy of the classifier. Therefore, we use multiple classifiers in a time-sequential fashion. Each classifier is specific for the time interval that has passed since the client's first attempt to connect, with respect to the parameter values and feature information it uses. The used parameter values per classifier are those obtained during the classifier training for the associated time interval, shown in Figure 7. The used feature information is also related to the time that has passed since the first

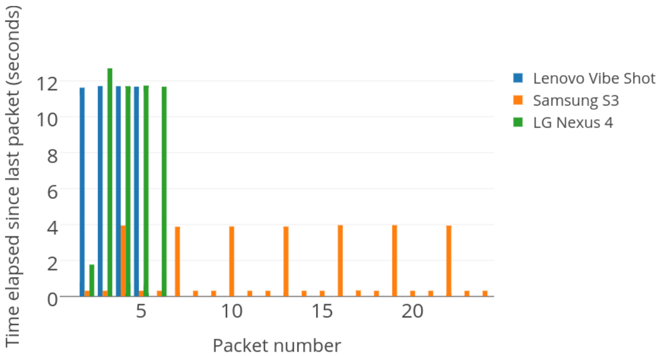

Fig. 13. Packets resent when there is no reply to the Authentication Request

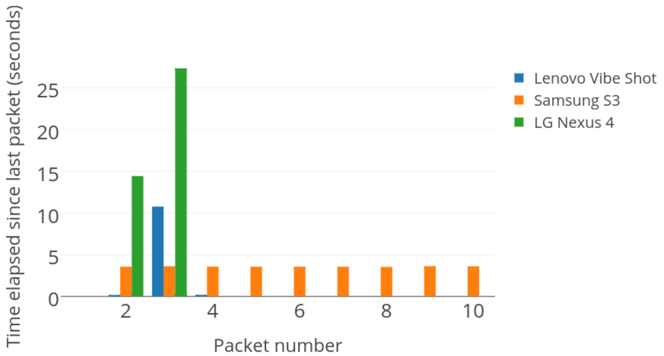

Fig. 14. Packets resent when the Authentication Request is denied

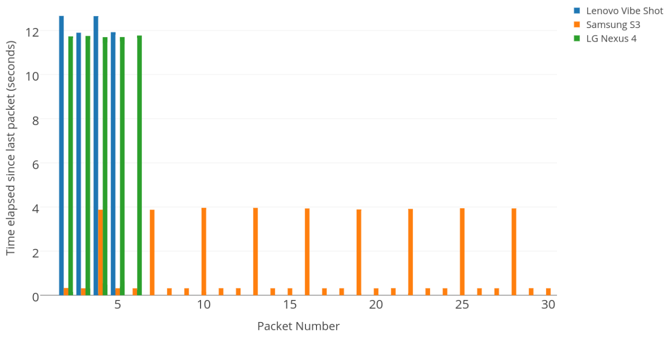

Fig. 15. Packets resent when there is no reply to the Association Request

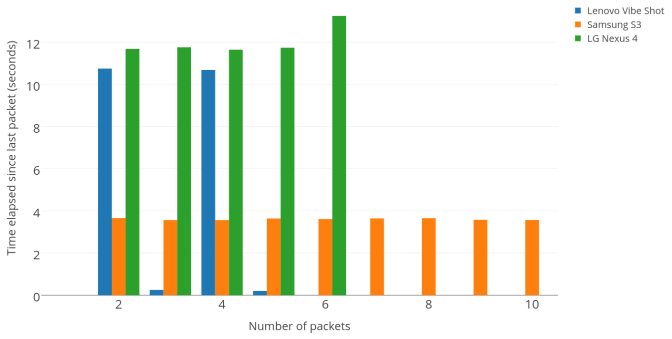

Fig. 16. Packets resent when the Association Request is denied

connection attempt: if it is the first attempt, the instantaneous value of bus speed is used; for subsequent attempts, the speed average since the first attempt is used.

There are two additional mechanisms to cover practical situations.

Monitored list tracks devices that tried to connect previously and were not granted connection, by keeping MAC addresses, time of the first connection attempt and the bus speed 10 seconds before the first time seen.

White list tracks devices are being classified as outside and allows them to connect after a certain time N_{wl}, as a bypass mechanism to handle false negatives – situations in which a user device is actually inside the bus but classified as being outside.[1]

C. Operation and Implementation

We detail the overall operation and order of actuation of the mechanisms discussed above. Figure 18 presents the overall workflow of the algorithm. Upon arrival of a DHCP probe from the network, the algorithm executes the following steps:

1) Bypass mechanism: Check if client has been trying to connect for a longer time than N_{wl}.

- If so, it is moved to the *White list* and granted connection.
- If not, the algorithm follows to the next step.

2) Classifiers: Check if client has tried to connect previously, by searching its address in *Monitored list.*

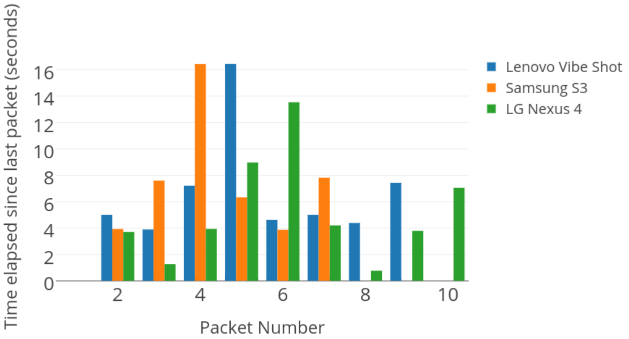

Fig. 17. Packets resent when the DHCP packets are ignored.

- If not (i.e. it is the client's first attempt), the classifier decides using the instantaneous bus speed as feature, and the parameter values for the start of the connection (from Figure 7). The process is shown in Figure 19.
- If so, depending on whether client is trying to connect at intervals (7.5,12,5], (12.5,17.5] or (17.5,22.5] seconds, the classifier uses the parameters corresponding to the 10, 15 or 20 seconds intervals (refer to Figure 7 for values) and the speed averages for those periods as input feature. The process is shown in Figure 20.

The gatekeeper mechanism implementation was broken down into two software modules. The core algorithm was incorporated in the *dnsmasq* process code. This includes the collection of classifiers and the bypass mechanism (that enforces immediate connection) for devices trying to connect

[1] Currently, the time threshold N_{wl} is set to 30 seconds.

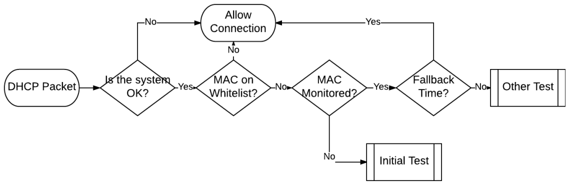

Fig. 18. Gatekeeper algorithm upon arrival of a DCHP probe.

for more than N_{wl} seconds. A parallel process called Bus Monitor was created to off-load non-essential functionalities, keeping code changes in the DHCP server to a minimum. The Bus Monitor process stores the last 20 seconds of the speed, which allows *dnsmasq* to calculate the mean of the bus speed and feed it to the classifier.

VI. Conclusions and Future Work

In this work we evaluate the feasibility of a mobile access point detecting the relative position of a mobile device. We collected a large-scale dataset of mobility and AP connection traces from a bus fleet equipped with WiFi service. A decision tree classifier was created with features based on the bus speed and RSSI values at different intervals with respect to the connection instant. We observed that RSSI was deemed irrelevant, whereas bus speed was an important feature. True positives rates increase as the time interval between connection start and classification instant increases. We proposed a gatekeeper mechanism to avoid connections from mobile devices that are outside the bus based on the developed classifier. A implementation proposal of this mechanism in a widely-used open-source WiFi network stack is presented. Despite the trade-off between Internet connection delay and classifier accuracy, our solution smooths the experience of the users outside the bus. Moreover, when a user boards the bus, he or she is likely to be waiting in the queue to enter the bus, validate the transportation ticket and find a seat, covering some of the time until Internet becomes available.

For further improvement, additional data could be collected from the city buses, and bus GPS traces could be used to learn the bus speed prior to the connection start. The gatekeeper mechanism could be deployed in a few mobile APs of the bus fleet for a test run, with measures taken to record the users' ground truth – whether would they are outside or inside at connection time. During the data analysis, we observed that certain geographical areas of the city have a larger density of connection events. Some geographical criterion could be used to further improve the mechanism performance.

VII. Acknowledgements

We thank Veniam (https://veniam.com/) for discussions and data.

This work was funded by projects S2MovingCity (CMUP-ERI/TIC/0010/2014) and SmartCityMules (UID/EEA/50008/2013), and by applicable financial framework of R&D Unit 50008 (FCT/MEC through national funds and FEDER-PT2020 partnership agreement when applicable).

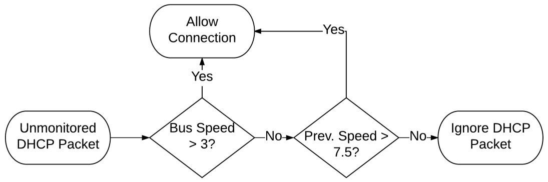

Fig. 19. Classifier algorithm for first connection attempts.

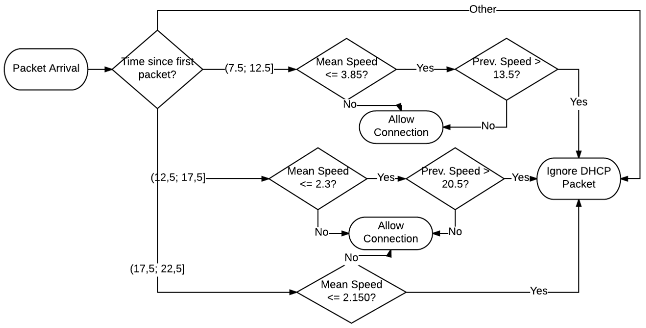

Fig. 20. Classifier algorithm for subsequent connection attempts.

References

[1] R. M. B. Reza Zekavat, *Handbook of Position Location:Theory, Practice and Advances*, 1st ed. O'Reilly, 2012.

[2] A. T. Parameswaran, M. I. Husain, S. Upadhyaya *et al.*, "Is rssi a reliable parameter in sensor localization algorithms: An experimental study," *Field Failure Data Analysis Workshop (F2DA09)*, 2009.

[3] K. Heurtefeux and F. Valois, "Is rssi a good choice for localization in wireless sensor network?" in *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*, March 2012, pp. 732–739.

[4] J. Krumm and E. Horvitz, "Locadio: inferring motion and location from wi-fi signal strengths," in *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, Aug 2004, pp. 4–13.

[5] A. Boukerche, H. A. Oliveira, E. F. Nakamura, and A. A. Loureiro, "Vehicular ad hoc networks: A new challenge for localization-based systems," *Computer Communications*, vol. 31, no. 12, pp. 2838 – 2849, 2008, mobility Protocols for ITS/VANET. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S014036640700518X

[6] R. Parker and S. Valaee, "Vehicle localization in vehicular networks," in *IEEE Vehicular Technology Conference*, Sept 2006, pp. 1–5.

[7] ——, "Vehicular node localization using received-signal-strength indicator," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3371–3380, Nov 2007.

[8] A. P. Subramanian, P. Deshpande, J. Gao, and S. R. Das, "Drive-by localization of roadside wifi networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008.

[9] B. Kihei, J. A. Copeland, and Y. Chang, "Predicting car collisions using rssi," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–7.

[10] K. Dhondge, S. Song, B. Y. Choi, and H. Park, "Wifihonk: Smartphone-based beacon stuffed wifi car2x-communication system for vulnerable road user safety," in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, May 2014, pp. 1–5.

[11] X. Wu, R. Miucic, S. Yang, S. Al-Stouhi, J. Misener, S. Bai, and W. h. Chan, "Cars talk to phones: A dsrc based vehicle-pedestrian safety system," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, Sept 2014, pp. 1–7.

[12] RapidMiner (company), "RapidMiner - Open Source Data Science Platform," 2016, https://rapidminer.com.

[13] G. Lui, T. Gallagher, B. Li, A. G. Dempster, and C. Rizos, "Differences in rssi readings made by different wi-fi chipsets: A limitation of wlan localization," in *2011 International Conference on Localization and GNSS (ICL-GNSS)*, June 2011, pp. 53–57.

[14] A. R. R. Franco, "Wifi roaming along urban routes," Master's thesis, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal, 2014.