Computer Communications 121 (2018) 71–82

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Context classifier for position-based user association control in vehicular hotspots

Pedro M. Santos[*,a], Leonid Kholkine[a], André Cardote[b], Ana Aguiar[a,c]

[a] Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias s/n, Porto 4200-465, Portugal
[b] VENIAM, Rua dos Heróis e Mártires de Angola 59, 4th floor, Porto 4000-285, Portugal
[c] Instituto de Telecomunicações, Rua Dr. Roberto Frias s/n, Porto 4200-465, Portugal

ARTICLE INFO

Keywords:
Association control
Mobile device
Vehicular hotspot
Contextual classification

ABSTRACT

Unintentional associations of mobile devices to on-board WiFi access points (APs) can affect the outdoor Internet experience of mobile device users, as the on-going cellular connection is broken and a short-lived WiFi connection is initiated. This disruption of the user experience can be avoided if the on-board AP learns whether the user device is inside or outside the bus and decides to accept its connection request or not. In this article, we present a classifier-based mechanism for on-board APs that accepts or denies user device associations based on a classification of the relative position of the device. An analysis of the problem in terms of connection duration and RSSI is presented to motivate the selected approach. We then describe a classifier to identify the user relative position trained on features extracted from contextual information. The classifier was trained with a large dataset of real-world WiFi-usage and mobility patterns of a public bus fleet from Porto, Portugal. The training procedure indicated bus speed as the most relevant feature, and that the RSSI measured at the on-board AP does not contribute. Finally, we propose a mechanism that grants or denies connection access to users based on the classifier output. We discuss how to integrate this mechanism in the AP network stack and evaluate its performance in real-world tests. Our solution can avoid 40% of the associations from users outside of the bus.

1. Introduction

Hotspots with WiFi access points (AP) and a connection to the cloud (via 3G or DSRC) are becoming common in public transportation fleets to provide Internet service to passengers. For ease of connection over multiple occasions, the on-board APs advertise the same Service Set Identifier (SSID) in all vehicles. In addition, many users of mobile devices tend to leave multiple wireless interfaces active when they are on the streets, specially WiFi and cellular. In case a user has already connected to the WiFi network of a bus, the network will be memorized 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 with the on-board AP, that may last for a very short period of time if the AP becomes out of range. Consequently, the latter connection is bound to cause a bad experience to the user as seamless Internet access is disrupted. Additionally, the association process also drains AP resources into a connection that will be inconsequential, thus potentially deteriorating the overall quality-of-service of the AP. As such, it is an *undesired* connection or association. Users to whom such accidental connections occur are typically also customers of the bus service (as they need to have used it once to have the network registered on their device), and higher customer satisfaction can be achieved if customers do not have their outdoor Internet experience disturbed by occasional buses passing by.

The undesired connection can be avoided if the on-board AP can learn whether the user is inside or outside the bus, and use that information to decide on whether to accept its connection request or not. There are several strategies to identify the user's relative position (if the user is inside or outside), but we seek a solution that is fully contained in the on-board AP. An alternative approach would be to request user input or action (e.g., logging in to a captive portal, installing a dedicated app). An AP-side solution has several benefits over this solution: (i) No human intervention is needed; (ii) the service provider needs not to rely on costumer awareness to achieve high adoption ratios of the solution; and (iii) the service provider can control the quality of the solution by rolling out new versions of the classifier and mechanism. Other non-AP-only options could pass by installing support hardware (e.g., Bluetooth beacons) or integrate with the ticket validation system, which would involve some deployment overhead and/or leave the

* Corresponding author.
E-mail addresses: pmsantos@fe.up.pt, pedro.miguel.santos@fe.up.pt (P.M. Santos), leonid.kholkine@fe.up.pt (L. Kholkine), acardote@veniam.pt (A. Cardote), anaa@fe.up.pt (A. Aguiar).

https://doi.org/10.1016/j.comcom.2018.03.004
Received 30 April 2017; Received in revised form 3 March 2018; Accepted 7 March 2018
Available online 13 March 2018

service provider dependent on third parties.

In this paper, we propose a classifier-based scheme to support the association decision at the on-board AP using available contextual information. We first describe the problem with field experiments designed to characterize these connections and that motivate the need of a classifier-based approach. The classifier was trained using a real-world dataset of mobility and WiFi connection traces collected at buses equipped with on-board APs and GPS. The automated training showed that the most relevant contextual information is the instantaneous speed and speed average over last 10 s. Finally, we also propose a design for a decision algorithm that incorporates the developed classifier and discuss some options of its implementation in software and integration in the network stack. An experimental evaluation of the solution performance with respect to blocking outside users is presented: The classifier manages to block about 40% of external users at the instant of the first contact. In practice, this means that 40% of outside users are prevented of engaging in an undesired connection, thus not having their outdoor Internet experience affected; and that the AP can avoid 40% of inconsequential connection requests, thus freeing resources for valid connections.

Our contributions are:

- Characterization of undesired connections and the conditions in which they are likely to happen;
- Development of a classifier for predicting the relative position of the on-board AP and a user device using real-world datasets;
- Design and implementation of an association decision mechanism in an open network stack, and real-world experimental performance evaluation.

The remainder of this article is as follows. In Section 2, we outline the existing state of the art. A characterization of undesired connections is discussed in Section 3. The development of a classifier to detect the relative position of the user is explained in Section 4. The design of a mechanism to identify relative position and decide on connection acceptance or refusal is presented in Section 5. The experimental evaluation of the mechanism performance is shown in Section 6. Final remarks and future work are laid out in Section 7.

2. Related work

We review the existing literature on decision schemes for association and handover, and solutions for relative position identification. Recent works on association in vehicular scenarios focus on service to vehicular users by infrastructural access points. The work of Xie et al. [1] presents an algorithm to improve long-term service duration by infrastructural WLANs. The authors of [2] propose a load balancing-aware scheme to decide association of user devices to heterogeneous cellular base stations. In [3], an algorithm framework that provides analytical performance guarantees in scenarios of multi-tier multi-cell environments is presented. Handover decisions schemes extend association schemes by considering additional criteria such as quality-of-service and/or logistics of sustaining on-going sessions. Given its close relation to association and wide body of literature, we review also the relevant works in this field. A taxonomy of handover decision schemes can be found in [4], with the proposed categories being RSS-based, QoS-based, Decision Function, Network Intelligence and Contextual. Our approach identifies with the later class; and within these, it sits among the decision mechanism that harness mobility prediction. In [5] the authors propose a decision scheme for handovers between WiFi and WiMax networks that takes into account the user's speed. For high user speeds, the authors defend that the handovers from WiMax networks to WiFi should not be easily triggered due to the WiFi base station's smaller coverage. The authors of [6] propose a handover decision scheme between WLAN and WWAN based on user micro-mobility prediction. In [7], a network selection mechanism for LTE user devices is presented: it seeks the best network (LTE or WiFi) to support application QoS requirements, using external user mobility prediction services. Works addressing the scenario of association/handover to in-vehicle networks typically focus on QoS issues. In [8], a mobility-aware call admission control (CAC) algorithm is proposed: when a hotspot-equipped bus stops to let passengers in, WLAN guard channels are reserved to support handover sessions from users coming in. In [9] a hybrid interworking scheme is proposed to support seamless vertical handover of IP sessions for vehicular passengers. We found no proposals of association or handover decision schemes for our target scenario.

Solutions for learning the relative position or distance between nodes fall into two classes: those that assume active participation from both nodes, or those in which a single node infers it. In the first case, nodes typically share a common communication technology. WifiHonk [10] is a vehicle-to-pedestrian (V2P) WiFi-based mechanism to avoid collisions, in which vehicular users advertise their positions via beacons. The authors of [11] describe an implementation of a DSRC stack in a smart phone-grade WiFi chip. Both solutions source GPS to obtain the nodes' position estimates; if the actual distance between nodes is close to the GPS receiver's position error, the computed relative distance can suffer from a substantial error [12]. Solutions that try to identify another node's position (or distance to it) passively are mostly based on RSSI. RSSI-based methods for localization include lateration methods, machine learning classification, probabilistic approaches and statistical supervised learning techniques [13]. For static users, the work of Krumm and Horvitz [14] tries to infer user motion and location from WiFi received signal strengths. The authors of [15] use the RSSI of V2V messages to predict vehicle collisions. The work of Parker and Valaee [16] proposes a collaborative RSSI-based localization solution. The use of infrastructural nodes is proposed in [17], in which the authors use RSSI and angle-of-arrival from infrastructural APs to improve their vehicle's position estimate. However, the studies of Parameswaran et al. [18] and Heurtefeux and Valois [19] show that, even if the target node is static, measured RSSI is not consistent enough through multiple measurements sessions to support reliable ranging/localization.

The topic of decision schemes for in-vehicle network associations (or handovers from urban WLAN/WWANs) has not been explored in detail, to the best of our knowledge. RSS-based localization solutions are a natural approach to explore, but existing solutions require specific software and/or hardware. Our solution abstracts from this shortcoming by being designed to operate on the AP side and sourcing contextual information available to the vehicle. In addition, our proposed scheme protects the QoS of the users that rest *outside* of the vehicle (an aspect that we have not seen explicitly addressed in literature), and harnesses a real-world dataset with a large number of users to develop a generic and universal solution.

3. Undesired connections to on-board access points

We present an introductory analysis to the problem of undesired connections between user devices and on-board access points (APs). Our definition of an undesired connection is as follows: a connection established between a stationary or slow-moving user device (smart phone) that stands by the road side and an on-board AP that passes by or is stopped for a short period near the user.

We developed and conducted two experiments to characterize connections between a user device and on-board APs. The experiments address the following cases: (i) The user device is by the road-side and the vehicle passes by; and (ii) the user device enters the vehicle. The characterization is made over time in terms of: (i) RSSI throughout connection; and (ii) duration of connection and connection stages (particularly in the first experiment). In performing these experiments, we expect to identify metrics and/or process behaviours that may indicate whether a user is outside or not with some degree of certainty. We detail next the methodology and results of both sets of experiments.

3.1. Road-side user

We enacted a scenario in which undesired connections are likely to occur, specifically that of a user standing by the road-side and a vehicle equipped with an on-board AP. In order to introduce the least variability to the RSSI samples and connection stages to be measured, the experiments were made in a simplified scenario, i.e., a personal car was used instead of bus, we selected a street with little traffic, and there were few nearby infrastructural WiFi APs.

3.1.1. Methodology

We equipped a car with an access point similar to those used in the public buses. The access point has an integrated GPS module, and ran software to collect the user device's RSSI. We note that the on-board AP installation in a private car presented differences to the setup in a bus. In the car, the on-board AP was placed on top of the dashboard, whereas in buses the APs are installed in a compartment just on top of the driver. Thus, the two setups enjoyed different heights to the ground and had different materials surrounding them.

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

- Timestamp of established connection and connection loss on the user device;
- Timestamp of the first and last transmitted ping;
- Sets of RSSI samples and GPS coordinates taken simultaneously at a rate of 1 second.

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 time between the last ping received and the instant the user device considers it no longer has connection to the AP (referred to as the "No signal but available" state) was computed *a posteriori* (i.e., there was no on-line evaluation of rates of responsive pings).

3.1.2. Results

The results were as follows:

1) RSSI throughout connection: Fig. 1 presents the RSSI measured at the on-board AP as the vehicle passes by the road-side user in the four runs. We also observe that there is no consistency across runs of RSSI values and distance between terminals at the instant the connection starts.

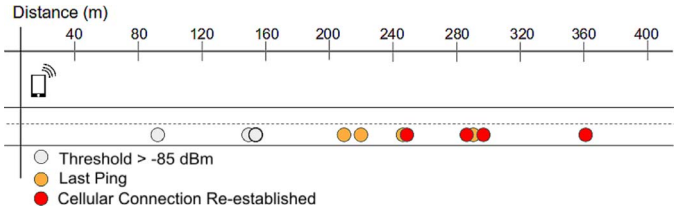

Fig. 2. Distance from the user when connection is lost.

2) Duration of connection and connection stages: In total, the duration of the connections during the tests lasted from 32 s to 64 s. We observed that, as the wireless signal starts fading, the connection from the point of view of the user goes through several stages until full disconnection from AP occurs:

- *Connected* - period during which RSSI value is above −85 dBm (less than 75 m away);
- *Weak signal* - period during which data transfers are possible but RSSI is below −85 dBm (more than 75 m away);
- *No signal but available* - period during which, although no connection is possible to the AP (verified through irresponsive pings), the user device still considers the connection as valid.

The several stages occur sequentially; in Fig. 2, we show the evolution of the disconnection process from the perspective of the user device as the vehicle moves away. The boxplots of time spent in each stage, for all four runs, are in Fig. 3.

The results show that the user device takes some time before concluding that the on-board AP is not in range anymore. This shows that, if a stationary user device connects to a moving AP, there is a period during which the device is mistakenly trying to communicate with the AP. One important behaviour observed on the user device is that the connection does not switch from cellular to WiFi until the device acquires an IP address from the AP. From the user device perspective, this means the cellular connection is not disrupted until the WLAN's DHCP process is complete.

3.2. Passenger user

In the second experiment, we took RSSI measurements inside buses. Unlike the first set of experiments, these experiments were carried out in a real-world setting. For a user riding the bus and sitting in a fixed location, we expect little variation of the relative distance, connection status and RSSI distribution over time (which were some of the relevant aspects to explore in the "Road-side User" experiments). Furthermore, there are additional aspects that can only be studied in a bus, such as

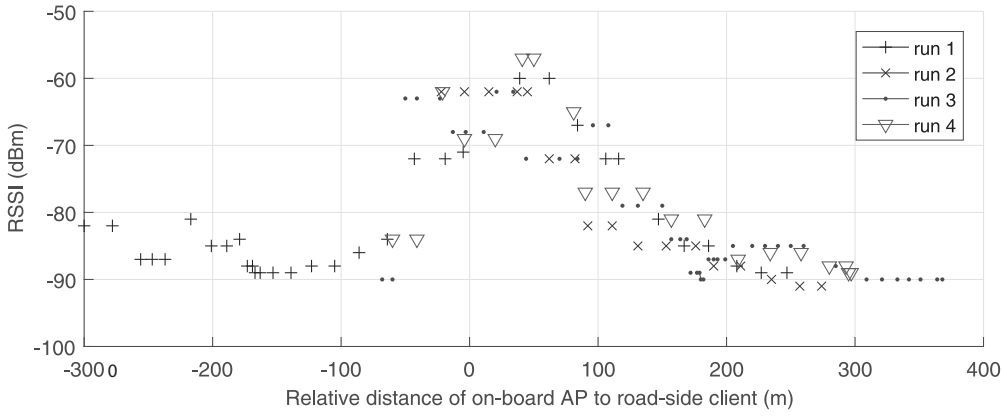

Fig. 1. RSSI profile as vehicle stops and user enters.

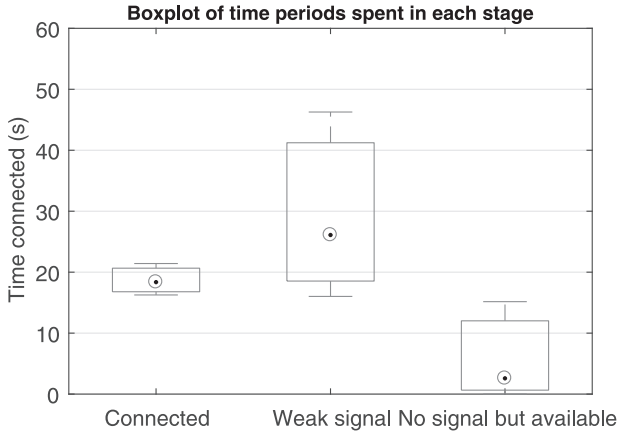

Fig. 3. Duration of connection stages.

the impact of sitting in different locations of the bus.

Due to the on-going WiFi service operation, the AP could not be used to record the RSSI of the user devices and, as such, the RSSI samples were taken on the client side, with Android devices using a customized application.

3.2.1. Methodology

The RSSI values were collected in a normal and an articulated bus, at different locations inside the buses. We performed 10 measurement sessions in each bus type, using the same Android device, and with a different number of passengers. Each session lasted for 30 s while sending pings to the AP. The RSSI was recorded every second.

3.2.2. Results

The results are shown in Fig. 4. We observe that the RSSI values measured vary with the user position inside the bus (as would be expected) and, for each location, differ between the regular and articulated bus (as distances are larger in the articulated bus). The RSSI values measured at a given location fit within reasonably well-defined ranges: in both types of buses, 50% of the RSSI samples are within a 4 to 6 dBm range for most locations (the only exception is the front location in articulated buses). We also recorded a slight difference between the RSSI measured at the front location in the two types of buses. We hypothesize that circumstantial factors are responsible for it, such as differences in makes and models of the vehicles, location and antenna orientation of the OBU setup in the vehicles, and of the user device during measurements.

3.3. Discussion and conclusion

We have characterized undesired connections in terms of RSSI and connection duration for the cases of: (i) User stands by the road-side while a vehicle equipped with an on-board AP passes by; and (ii) user is a passenger of the bus. In the first case, we observed inconsistency of the RSSI value taken at the start of the connection, distance to vehicle at connection establishment, and in the duration of the connection. In the second experiment, we observed that the RSSI values measured inside the bus, at different locations, meet well-defined values range depending of the location.

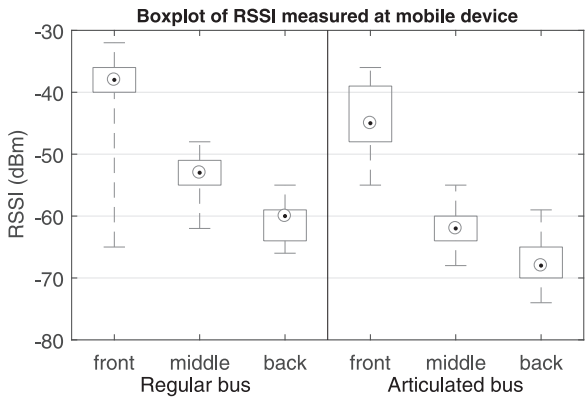

Fig. 4. RSSI measured inside the bus.

The first experimental session (“Road-side User”) was carried in a simplified scenario, i.e., without a significant number of the factors and phenomena that occur in a real-world scenario such as the existence of nearby WiFi networks, traffic lights and bus stops, jerky or unpredictable movements of the vehicles, user mobility and device attitude variations, among others. Despite the simplified scenario, we observed considerable inconsistency in the connection behaviour (e.g., RSSI and distance of first contact), which led us to conclude that a solution based on instantaneous or windowed analysis of RSSI alone would not be effective. This caused us to change our approach and pursue a strategy based on the analysis of large dataset from a real-world scenario (that implicitly captures the range of variables occurring in the real-world) and use machine learning to develop a classification solution (described in the following section). Nevertheless, we still included the RSSI of user devices measured at the AP in this dataset, motivated by the promising results of the second set of experiments, that was conducted in a real-world environment and thus already captured part of those factors and phenomena.

4. Context classifier of relative position

We now describe a classifier trained to identify the relative location of an on-board AP and a user device. The features used for online classification should be extracted only from information available to the on-board AP. Thus, for the purpose of training the classifier, we sourced a large-scale dataset of bus GPS traces and RSSI samples of user devices connected to the on-board APs. An initial set of features was constructed from this dataset based on the conclusions of the preliminary experiments, and we used the RapidMiner tool [20] for feature selection and classifier training.

For the remainder of this section, we present the used dataset, the set of initial features, the trained decision tree and an evaluation of its performance, and a discussion on the resulting classifier.

4.1. 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 user devices and on-board APs was recorded during one week on seven city buses. The buses did not stay on the same route from one day to the next; routes were assigned randomly. The following information was stored per packet received at the on-board AP: timestamp, MAC address, average RSSI value, GPS position, and GPS speed. Two packets with the same MAC address were part of the same connection if they had no more than 2 s of difference between timestamps. In case there was a larger difference, packets were assigned to separate connections. In total, around 5 million lines of data were gathered, corresponding to 14,063 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 12,040 connections.

The dataset did not have any ground truth annotation for our classification task – whether the connecting user device is outside or inside the bus. We performed artificial annotation of the dataset using two connection characteristics, the connection duration and the distance traveled by the bus during the connection, working under the assumption that connections that last long and during which the bus traveled a large distance cannot be caused by devices connected outside

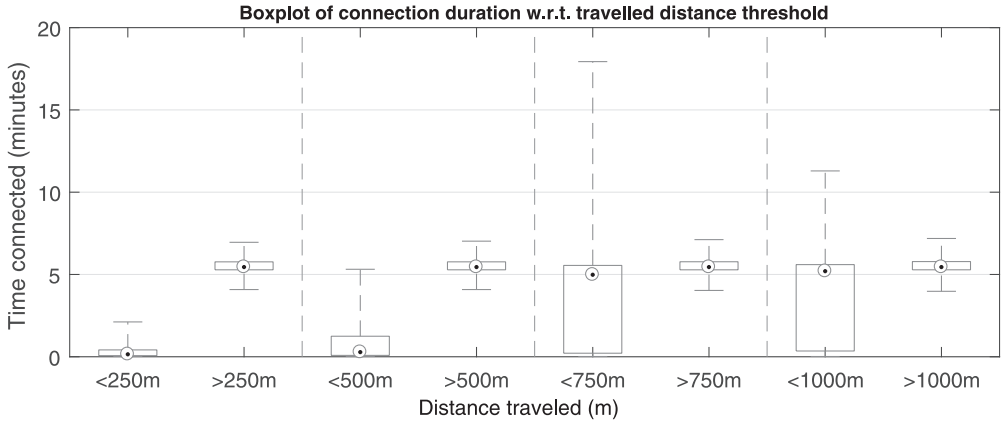

Fig. 5. Data segmentation, inside vs outside the bus, for distances with outliers removed using the Hample identifier.

the bus. We searched for discriminating criteria by binning connections into two classes with regards to a threshold of the distance traveled by the bus during the connection, and examining the distribution of the connection duration per class. We considered four arbitrary thresholds for the distance traveled by the bus – 250, 500, 750 and 1000 m. For each category, we removed outliers with respect to connection duration using the Hampel filter, a window-based method that substitutes samples more than three standard deviations away from the window median by that median. The results for each threshold are presented in Fig. 5. The 250 m threshold produces the best separation between the two artificially-annotated groups of connections: there is a clear distinction between brief and short connections (outside the bus) and long-lasting connections (inside the bus). Thus, we annotated the dataset according to the following criteria:

- *Outside*: connections during which the bus traveled less than 250 m, and connection lasted between 20 s and approximately 2 min;
- *Inside*: connections during which the bus traveled more than 250 m, and connection lasted between approximately 4 and 7 min.

We obtained up to 2,350 connections outside the bus and 7,400 connections inside the bus.

During the data analysis, the connections outside the bus were mapped geographically. The result is displayed in Fig. 6, and it can be observed that certain geographical areas of the city have a larger density of connection events.

4.2. Feature selection

An initial set of features were produced from the dataset for input to the classifier training procedure. The features could be divided into two classes – RSSI-based and speed-based. We selected the RSSI and speed at connection time, and 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 on-board AP: the longer the time interval to decide whether the user is inside or outside the bus, the longer the user must wait before being given Internet access. The bus speed prior to the connection start was also included as it may indicate whether the bus had just been at a stop. This speed prior to the connection start was computed as an average of the N_{prev} seconds prior to the connection. Since we only collected the speed during the connection, this feature was extracted from other ongoing connections on the same bus in the relevant time window.

Overall, the features we generated were the following:

- RSSI at connection time;
- Mean / median / std. dev. of the RSSI for the first N seconds of connection;
- Mean / median / std. dev. of RSSI variation for the first N seconds of connection;
- Mean / median / std. dev. of the Exponential Moving Average (EMA) of the RSSI;
- Vehicle speed at connection time;
- Mean / std. dev. of the vehicle speed for the first N seconds of

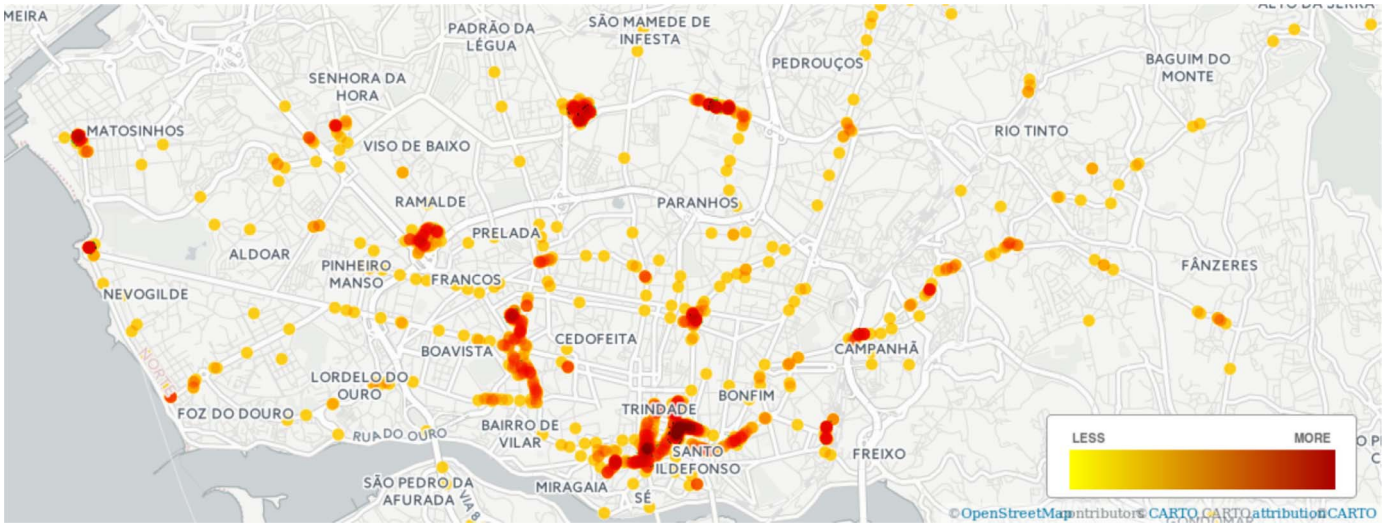

Fig. 6. Geographical distribution of the connections outside the bus.

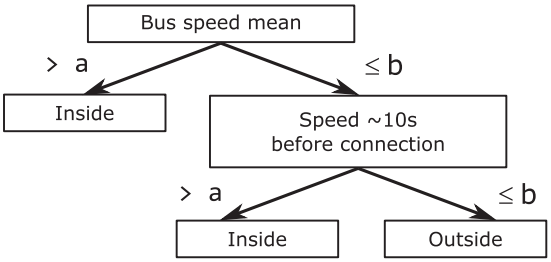

Fig. 7. Produced decision tree (see Table 1 for parameter values).

connection;
- Mean / std. dev. of the vehicle speed for the N_{prev} seconds prior to connection;
- Slope / intercept value of the linear regression of RSSI;

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

4.3. 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 final dataset of 7400 connections inside the bus was sub-sampled to obtain 2,350 connections, thus matching the number of outside connections. The connections were split into 70% for training the decision tree and 30% for testing. RapidMiner was run separately for $N = \{5s, 10s, 15s, 20s\}$.

The decision tree output by RapidMiner is shown in Fig. 7, with the produced parameters values for the different N being shown in Table 1. Respective rates of true positive (sensitivity), true negatives (specificity) and ratio of overall correct classifications (accuracy) are shown in Table 2. 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 4.4.

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 user 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 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 inference, 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 user device is inside or outside the bus. This feature provided considerable performance improvement. In Table 3, 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 implemented separate decisions trees for the various instants in a sequential fashion and ran the classification function in RapidMiner. The results for the percentage of the users that are allowed to connect or not is shown on the Fig. 8. We see that the percentage of the users that are inside the bus and can connect (true positive rate) can be increased by 10% at the cost of some connection delay (10 s) and a 4% increase in false positives.

Table 1
Decision tree parameters, applied in the decision tree of Fig. 7, with respect to elapsed time since connection start. Values in km/h.

Elapsed time	0 s	10 s	15 s	20 s
Parameter *a*	3.5	3.85	2.3	2.15
Parameter *b*	7.5	13.5	20.5	–

Table 2
Sensitivity, specificity and accuracy of the decision trees for different time intervals

	Elapsed time			
	0 s	10 s	15 s	20 s
Sensitivity (inside the bus)	73.59%	80.77%	83.85%	87.44%
Specificity (outside the bus)	63.85%	61.03%	56.92%	55.90%
Accuracy	68.72%	70.90%	70.38%	71.67

Table 3
Sensitivity, specificity and accuracy of the decision tree with and without previous speed information (at first contact instant and after 20 seconds)

	Elapsed time			
	0 s		20 s	
Prev. speed information	*Not used*	*Used*	*Not used*	*Used*
Sensitivity (inside the bus)	65.38%	73.59%	81.22%	87.44%
Specificity (outside the bus)	67.93%	63.85%	61.31%	55.90%
Accuracy	66.79%	68.72%	71.41%	71.67%

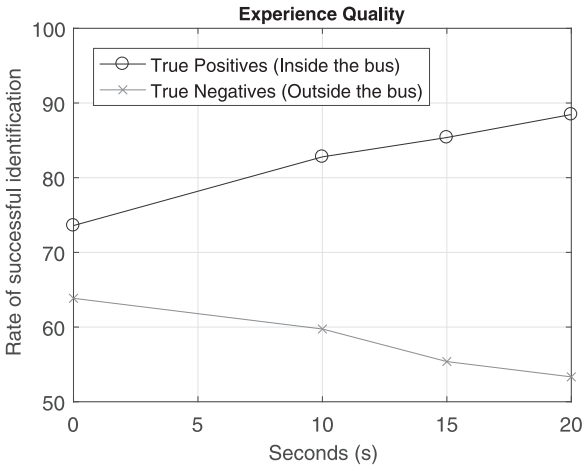

Fig. 8. True and false positives for different time intervals.

4.4. Why RSSI is not a good feature

We looked in detail at the RSSI samples of the dataset to understand the reason of not being selected as feature during the classifier training. We searched for differences in RSSI behaviour among connections outside and inside the bus, and for that we computed the mean, median and standard deviation during the first 20 s of connection. The RSSI mean and median can be visualized in Fig. 9. It is possible to see that there is little difference between the RSSI inside and outside the bus. Fig. 10 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.

This analysis further stresses the difficulty of using a RSSI-based metric to discriminate users inside and outside the bus, as observed in the initial characterization experiments (Section 3). In using a real-world large-scale dataset, we were made aware of another factor that

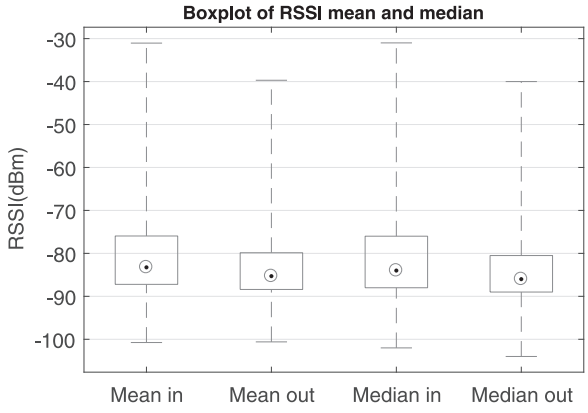

Fig. 9. RSSI mean and median for inside (*in*) and outside (*out*) connections.

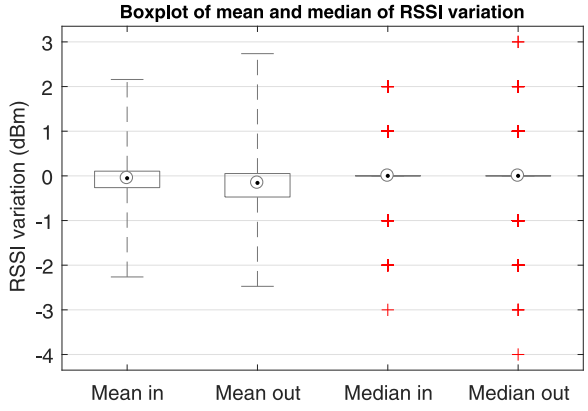

Fig. 10. Mean and median of the variation of RSSI.

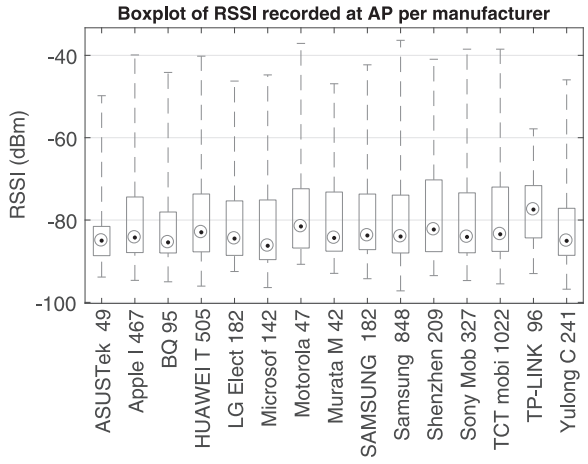

Fig. 11. Measured RSSI values per manufacturer.

may contribute for RSSI similarity. The range of user devices detected encompasses a variety of wireless transceiver hardware that, in turn, produce different RSSI levels in similar conditions, as shown in [21]. We were able to distinguish device makes and models of the dataset via the first half of the MAC address, that is assigned to the chipset manufacturer by the IEEE Standards Registration Authority (informally referred to as “assignment”). In this manner, we could plot the RSSI median by manufacturer, as seen in Fig. 11. Only the connections that traveled for over 1 km of distance were included in the plot, and only if there were more than 40 connections for the same assignment. The result supports our hypothesis that RSSI levels varies among chips of different manufacturers, contributing to the frailty of RSSI as a feature in this classification task.

5. System design and integration in network stack

We now describe a system design that incorporates the classifier into the on-board access points – the Gatekeeper mechanism. The Gatekeeper mechanism evaluates if a user device is inside or outside the bus before allowing connection, using the classifier described in the previous section.

In this section, we present a preliminary study meant to identify the best occasion in the connection setup procedure to deny association, the main considerations driving the design of Gatekeeper, and a description of the operation and implementation of the mechanism. An experimental real-world evaluation of Gatekeeper performance follows in the next section.

5.1. Preliminary study: Impact of connection denial

We sought to identify the best occasion, during the connection setup process, to refuse a connection. This will inform us about which network stack component should Gatekeeper be integrated in. Our criteria to identify the best stage is that that: (i) Causes the least disruption in the user device being denied; (ii) causes the most homogeneous response across different devices. For that purpose, we studied the behaviour of multiple user devices when denied a connection to an AP. The stages required for a user device to connect to a WiFi access point are four: Discovery, Authentication, Association and IP assignment. During Discovery stage, the user device searches for an AP actively or passively. Once an AP is found, the user device initiates two message exchanges leading to Authentication and Association with the AP. Regarding DHCP operation, the IP assignment process can take several seconds due to two reasons [22]: (i) If the devices find WiFi networks with a 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 3.1.2, the previous cellular connection is only broken when the user device has successfully acquired an IP address on the WiFi connection. Thus, the connection can be denied at the DHCP IP assignment stage or the stages that precede it, except Discovery (no denial primitive is available). In any stage, the AP can deny access either explicitly by sending a message, or implicitly by not replying.

On the user device side, the behaviour of a user device in either case for the three stages is not standardized. We designed a test to understand how various user devices behave when denied connection to an AP. The test was conducted on 3 smart phones 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 a modified version of *hostapd* that either did not reply to requests or denied them, to emulate the available options. For one minute starting from the first authentication packet received, the timing of the packets was recorded to extract the inter-packet interval.

The results can be seen in the Figs. 12–16. We observed that mobile devices can behave very differently among manufacturers. For example, when denied an Association request (shown in Fig. 13), the Samsung S3 sends a packet every 4 s, and the LG Nexus sends 3 packets with a 30 s interval between the second and third packets.

From this study, we conclude that the DHCP stage is the preferential stage to deny the connection, as the behaviour exhibited by the different mobile devices is most similar. As seen in Fig. 16, ignoring DHCP packets causes fast retransmission in all devices (although inter-packet periods may grow to 16 s), and the inter-packet interval increases until a certain threshold is reached and then restarts from a lower value. As a final remark, we observe in all cases that the inter-packet interval is larger when denying a connection.

5.2. Gatekeeper design

The design of Gatekeeper followed three main requirements:

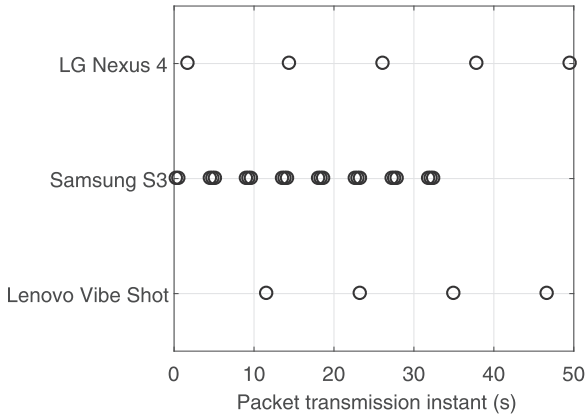

Fig. 12. Packets resent when there is no reply to the Authentication Request.

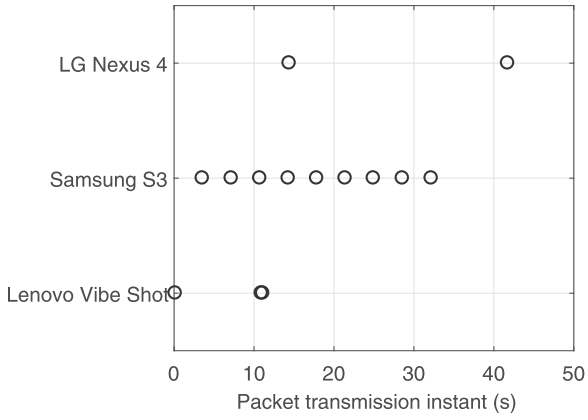

Fig. 13. Packets resent when the Authentication Request is denied.

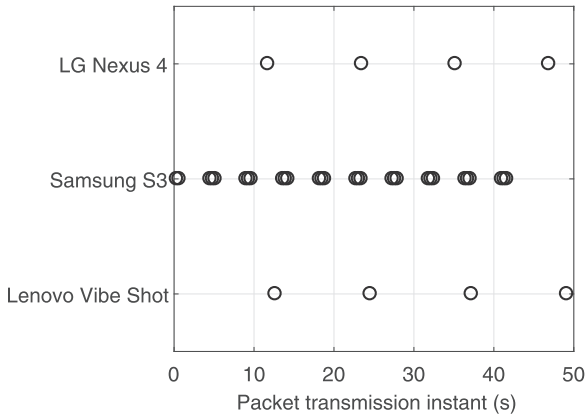

Fig. 14. Packets resent when there is no reply to the Association Request.

- Disrupt the denied user experience the least possible;
- Minimize the access time to inside users;
- Handle false negatives.

The first design requirement is meet by leveraging the previous conclusion that denying a connection is best at the DHCP stage. Thus, the Gatekeeper mechanism is implemented in the Linux kernel DHCP server code *dnsmasq*. This allows us to access directly the DCHP probes from user devices that arrive from the network.

The requirement that Gatekeeper must keep the delay experienced by users inside the bus to a minimum needs to be traded off with the fact that a longer period of data collection results in better accuracy of the classifier (as concluded in Section 4). Our approach is to issue

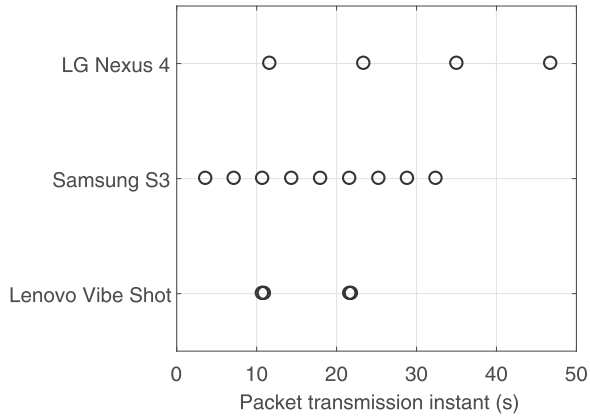

Fig. 15. Packets resent when the Association Request is denied.

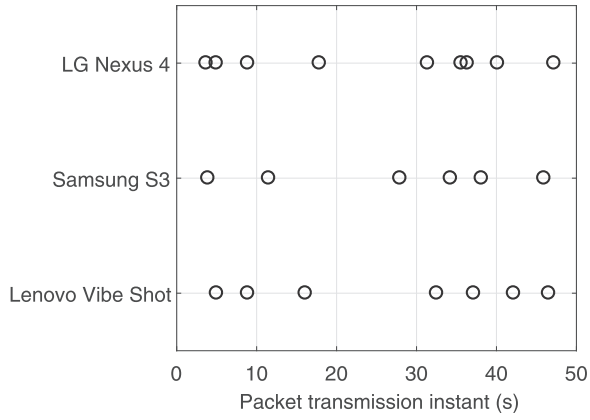

Fig. 16. Packets resent when the DHCP packets are ignored.

additional classifications at known intervals since the instant of the first connection attempt, in addition to the initial classification (at the instant of the first attempt) and while the user has not been granted access. As time elapses, the classifier parameter values and input feature information are updated to improve performance. For a given elapsed time interval, the used parameter values and input features are those obtained in classifier training for the associated time interval, shown in Fig. 7. Notably, the input feature information is updated as follows: in the first attempt, the instantaneous value of bus speed is used; in subsequent attempts, the speed average since the first attempt is used.

Finally, false negatives are situations in which a user device is inside the bus but recurrently classified as being outside. False negatives are handled with a bypass mechanism: after a timeout N_{wl}, these devices are allowed to associate to the AP.

There are two data structures to support the operation of the classifier and implement the end application goal:

- *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 classified as *outside* and flags them after the bypass timeout N_{wl} expires to allow their connection, in case they continue to try to access the AP.

5.3. Operation and implementation

We detail the overall operation and order of actuation of the mechanisms discussed above. Fig. 17 presents the overall workflow of the algorithm. Upon arrival of a DHCP probe from the network, the algorithm executes the following steps:

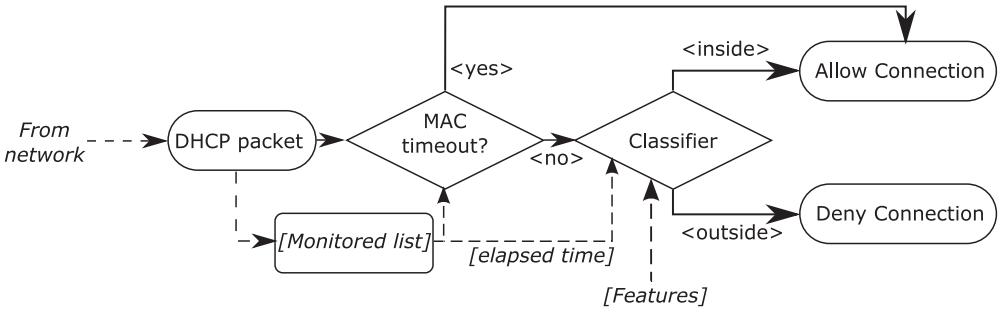

Fig. 17. Flowchart of Gatekeeper operation.

1) Apply bypass mechanism: Check if the user device 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) Apply classifiers: Check if user device has tried to connect previously, by searching its address in *Monitored list*, and apply the appropriate classifier mechanism:

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

The application rules and classifier parameter used in each of the previous cases are summarized in Table 4.

The Gatekeeper mechanism implementation was broken into two software modules: the core algorithm and the Bus Monitor process. The core algorithm includes the collection of classifiers and the bypass mechanism, and was incorporated in the *dnsmasq* process code. The Bus Monitor process runs in parallel with the core algorithm. It was created to off-load non-essential functionalities, thus keeping code changes in the DHCP server to a minimum. Bus Monitor stores the speed of the last 20 s, allowing *dnsmasq* to compute the mean of the bus speed and feed it to the classifier.

6. Experimental evaluation of gatekeeper solution

We conducted experiments to evaluate the performance of Gatekeeper in an urban scenario, using a private vehicle. We did not have the possibility of deploying Gatekeeper in a bus AP, and therefore we could only evaluate the performance of the classification regarding users that did not enter the vehicle. Nevertheless, this is the most relevant use-case as it is the one prone to undesired connections.

Table 4
Elapsed time-differentiation of parameters and inputs.

Time elapsed t	Threshold values		Input feature	
	Speed	Prior Speed	Speed	Prior Speed
0	3.5	7.5	Instantaneous speed	Speed avg. for $[N_{prior},0]$
(7.5, 12.5]s	3.85	13.5	Speed avg. for [0,t]	*idem*
(12.5, 17.5]s	2.3	20.5	*idem*	*idem*
(17.5, 22.5]s	2.15	–	*idem*	*idem*

6.1. Methodology and collected dataset

We installed an on-board access point in a private vehicle that advertised the same SSID as the public bus WiFi service. We performed a circuit through the city passing by some of the locations where the most connections from outside users occurred, as seen in Fig. 6. The following data was collected:

- Timestamp of first and subsequent DHCP packets from users, and associated classifier decisions;
- RSSI of connected devices and GPS coordinates at a rate of 1 s.

The ground truth of the relative position of users is known, as all users were outside the vehicle and would be not be entering. We set the timeout N_{wl} for the bypass mechanism at 25 s, as the classifier was only trained to handle DHCP stage durations of up to 20 s.

We carried out two measurement sessions of two and half hours, in separate weekdays and at rush hour (5:00pm–7:30pm) to maximize connection attempts. The users were anonymous pedestrians that happened to be on the street during the experiment and had no relationship whatsoever with the experiment team. The RSSI measurements were obtained at the on-board access point. In order to replicate a bus driving patterns, speed and acceleration were kept moderate. Furthermore, on occasion we would stop at bus stops and wait for an interval similar to that that a bus takes to off-load and up-load passengers. We removed connections in which the DHCP stage lasted more than 180 seconds and separated independent connections from the same MAC address if RSSI samples were apart by more than 300 seconds. In total, 180 connection attempts from different users were recorded.

6.2. Results

We now present operation and performance analyses of Gatekeeper drawn from real-world measurements. We address the accuracy of the classifier and access mechanism, the quality of the usability (e.g., access latency), and vehicle speed at first contact for comparison against the final Gatekeeper decision. As a final analysis, we evaluate if our criteria for artificial labeling of the ground truth dataset was valid or not.

6.2.1. Performance of classifier and gatekeeper

We differentiate the success rate of the classifier and the acceptance rate of the Gatekeeper for purposes of performance evaluation. Note that the decision of Gatekeeper follows the classifier output up to the bypass timeout, after which the classifier is overridden and access may be granted to a user even though the classifier indicates an *outside* classification.

Table 5 summarizes the two rates as time progresses (from left to right in the table). Upon the first DHCP packet reception, the classifier evaluated 71 out of 180 users as being *outside*, resulting in a success ratio of 40%. The remaining users were considered *inside* and granted immediate access by Gatekeeper. As time progresses to the bypass

Table 5
Classifier and Gatekeeper performance (read from left to right for time progression; total number of connections = 180).

Decision criteria	*Before timeout: Classifier output*			*After timeout: Bypass output*	
	Classification	First classif.	Up to timeout	*Action*	After timeout
Outcome of classifier/ mechanism (nr users per class)	Inside	109	+5		–
	Outside	71	−5	New DHCP packet	17
				No more packets	49
GateKeeper decision	Accepted	109	114	Accepted	131
	Denied	71	66	No follow-up	49
Classifier performance	Ratio correct	40%	37%		–

timeout (of 25 s), Gatekeeper issues new classifications for the some of the *outside* users based on changes in the classifier parameters and collected features. As visible in the table, the classifier eventually reverted its decision on 5 users initially classified as *outside* to being *inside*, and thus were granted access by Gatekeeper.

After the timeout, the bypass mechanism accepts users with an *outside* classification that are still sending packets. The number of such users was 17. The remaining 49 users classified as *outside* were never classified as inside nor sent a new packet after the timeout period.

6.2.2. Usability analysis

We evaluate the usability of our Gatekeeper solution by two metrics: (i) *Decision latency*: The interval since the first DHCP packet is received and the instant a decision is made at Gatekeeper; and (ii) *Access latency*: The interval since the first DHCP packet is received and the instant at which the user device is assigned an IP and Internet access is finally possible.

The decision latency is dependent on two factors: (i) Given that classifier parameters are updated as time passes by (see Table 4), the features may come to match the classifier criteria for an *inside* classification; and (ii) as discussed in Section 4.4, devices of different makes and models feature different behaviors regarding DHCP packet transmission timings. The CDF of the decision latency is shown in Fig. 18, for users classified initially as *outside*. If the interval is zero, the decision remains the same; otherwise, a revision of the decision occurred (in all cases from *denied* to *accepted*). We observe that the decision remains the same for 69% for users attempts classified as *outside* upon reception of the first DHCP packet. A number of users are granted access upon reception of subsequent DHCP packets within the interval up to the timeout of 25 s. For the remaining 23% of the users, the first decision based on the classifier output is overridden by the bypass mechanism after the 25 s timeout (depicted with the dash-dot line).

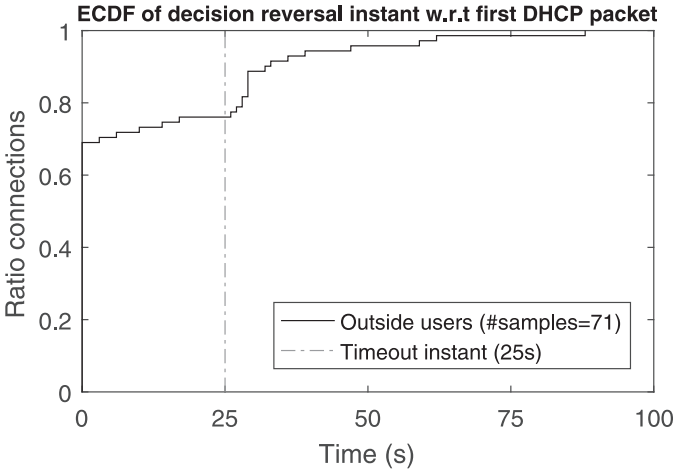

Fig. 18. Decision inversion latency for users initially classified as *outside*.

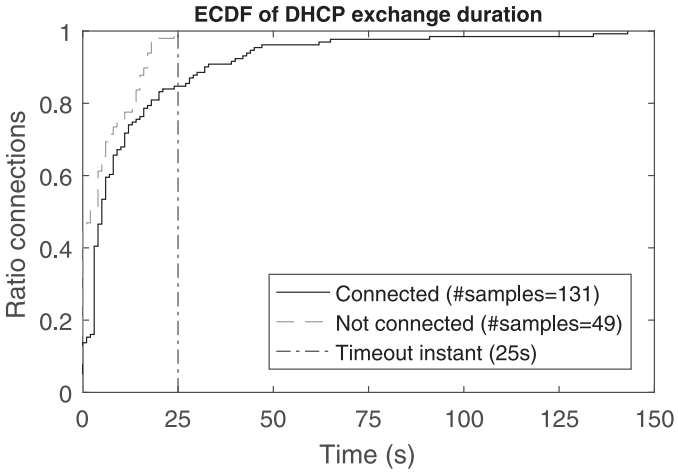

Fig. 19. DHCP stage duration and service latency for users.

The access latency encompasses the decision latency and the additional interval that it may take for the user device to receive an IP address. Due to the difference in DHCP request timing policies among devices of various makes and models, there is some variability about the instant at which the user device is assigned an IP even though the classifier has already issued internally an *accept* decision for that device. The CDF of the access latency is shown in Fig. 19. We note that, for example, among the 17% of users that connected after the bypass timeout (which are automatically accepted), there is considerable difference among the instants at which the DHCP exchange ends. This result shows how various users may register a different experience with Gatekeeper due to a blend of factors stemming from the user device and the Gatekeeper operation. This plot also confirms that the users that did not connect after the bypass timeout ended their interactions with the AP before the timeout, as expected.

6.2.3. Speed at first contact

We plot the speed at which the bus traveled at the instant of the first reception of a DHCP packet. The threshold speed for a decision at this instant is 3.5 km/h. Fig. 20 presents the CDF of these speeds for the users that were classified as *inside* or *outside* at the first contact. Two observations can be extracted: (i) for the *outside* users, 96% had less than 3.5 km/h as expected; (ii) for the *inside* users, 24% of the speeds at the first packet reception were below the threshold; the remaining connections show a wide range of speeds. Note that, for the latter case and regarding speeds below the threshold, the deciding criteria might have been the average speed of the 10 seconds prior to the first DHCP packet.

6.2.4. Validation of ground truth criteria

Finally, we evaluate the accuracy of our criteria to perform the ground truth annotation for the large-scale dataset used to train the

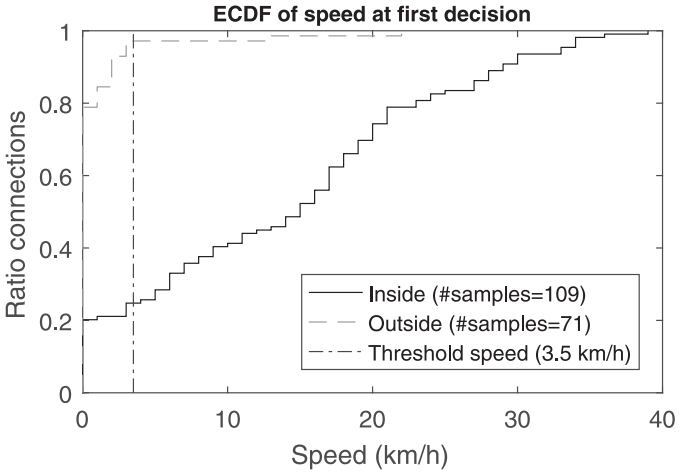

Fig. 20. Instantaneous speed at first contact instant per class.

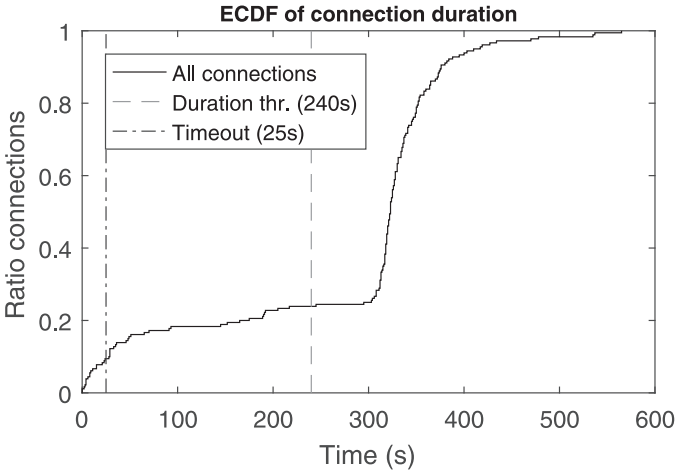

Fig. 22. Duration of connections.

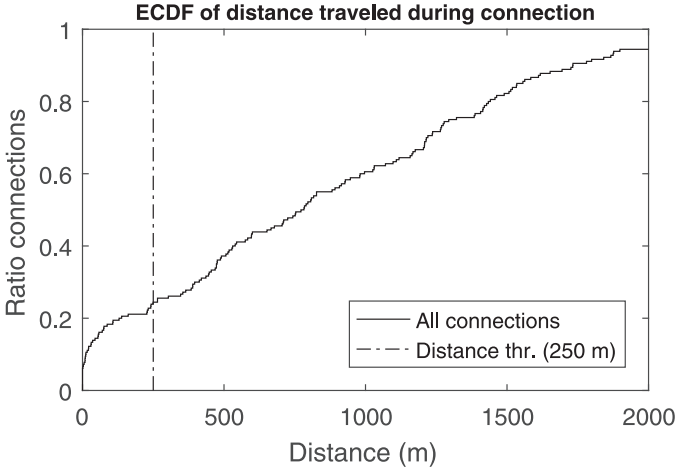

Fig. 21. Distance traveled during connections.

classifier. As discussed in Section 4.1, the criteria to artificially separate user devices into *outside* and *inside* was based on a threshold-based classification of distance traveled by the vehicle and connection duration, with thresholds of 250 m and 4 min (240 s) respectively. Given that in the current experiments the ground truth was available (all user devices were outside), we could evaluate if these criteria are observed. The CDF of distance traveled by the vehicle while the user device is associated to the AP is shown in Fig. 21. We observe that in only around 25% of the connections match our threshold for relative position identification of outside users. In Fig. 22 the CDF of the connection duration is shown, and we observe that only 23% of connections meet the criteria for being classified as outside.

6.3. Discussion

We observe that, in this experiment, the developed classifier had a success rate of 40% in identifying outside users at the first instant. Using a test dataset, a value of 63% for instantaneous decisions had been achieved, as shown in Fig. 8. The higher rate of *inside* identifications is related to the wide range of speeds the on-board AP experiences at the time of the first contact between AP and user device. The trained classifier assigns the classification of *outside* if the bus is traveling slower than 3.5 km/h. We observed in Fig. 20 that the vehicle speed upon reception of the first DHCP packet is, in 62% of the cases, superior to this threshold.

To explain this discrepancy, we hypothesize that our experiments may have not captured the full range of conditions and situations that buses experience everyday throughout the whole city. Additionally, the procedure used to annotate the initial dataset may be inaccurate with respect to the conditions of these experiments, which had a much more limited scope than the full dataset. The later aspect is corroborated by the observations of the previous section that only 25% of the connections feature a travel distance classifiable as *outside*, and likewise for 23% of the connections regarding duration. Notwithstanding, it is infeasible to obtain a perfect ground truth for this problem.

As main conclusion, Gatekeeper is shown to be able to reduce the amount of inconsequential connections, and thus the respective load on the mobile hotspot and disruption to outside users, by 40%.

7. Conclusions and future work

In this work we evaluate the feasibility of an on-board access point detecting the relative position of a user device, to support the decision on whether to allow the device to associate or not. This mechanism is designed to protect and smooth the outdoor Internet experience of bus WiFi network users when not riding the bus., This is achieved in two key ways: (i) As their cellular Internet experience is undisturbed by passing buses; and (ii) the need to manage interface operation at their mobile device is obviated. Initial field experiments showed that a solution based purely on RSSI might be impractical. Thus, we sourced a large-scale dataset of mobility and AP connection traces from a bus fleet equipped with WiFi service to train a decision tree classifier. Input features were based on the bus speed and RSSI values at different intervals with respect to the connection instant. After training, we observed that RSSI was deemed irrelevant whereas bus speed was an important feature. Finally, we proposed Gatekeeper, a mechanism based on the developed classifier to be incorporated in the network stack of embedded devices, such as the on-board AP. Gatekeeper adds features to provide a seamless experience to users that enter the vehicle: if not granted access immediately, Gatekeeper periodically revises the user device classification after the first contact, and grants unconditional access after a timeout of 20 s, a period similar in scale to the IP assignment stage. Field experiments in a private vehicle showed that the Gatekeeper classifier identified around 40% of the users that were outside at the first instant. In practical terms, this translates into 40% of outside users being denied association to the on-board AP; if these users had an on-going cellular connection, their Internet experience was not disrupted.

For further improvement, additional data could be collected with the actual user location, either via user input or integration with the ticket system, in order to create a dataset with a true ground truth. Regarding implementation, the Gatekeeper mechanism could be deployed in a few on-board APs of the bus fleet for a test run. Other

potential improvements may come from analyzing the time a user who is about to enter the bus takes before connecting to the hotspot – in waiting in the queue to enter the bus, validating the ticket and finding a seat –, in order to enhance the overall user experience with Gatekeeper. Finally, some geographical criterion based on Fig. 6 or on the location of bus stops and traffic lights could be used to further improve the mechanism performance.

Acknowledgements

We thank Veniam (https://veniam.com/) for discussions and data, and the reviewers and Damião Rodrigues for the insightful comments and suggestions that helped us improve the article. This work was funded by projects SmartCityMules (UID/EEA/50008/2013) and S2MovingCity (CMUP-ERI/TIC/0010/2014), and by applicable financial framework of R&D Unit 50008 (FCT/MEC through national funds and FEDER-PT2020 partnership agreement when applicable).

References

[1] L. Xie, Q. Li, W. Mao, J. Wu, D. Chen, Association control for vehicular wifi access: pursuing efficiency and fairness, IEEE Trans. Parallel Distrib. Syst. 22 (8) (2011) 1323–1331, http://dx.doi.org/10.1109/TPDS.2011.17.

[2] Z. Li, C. Wang, C.J. Jiang, User association for load balancing in vehicular networks: an online reinforcement learning approach, IEEE Trans. Intell. Transp. Syst. 18 (8) (2017) 2217–2228, http://dx.doi.org/10.1109/TITS.2017.2709462.

[3] W.C. Ao, K. Psounis, Approximation algorithms for online user association in multi-tier multi-cell mobile networks, IEEE/ACM Trans. Netw. 25 (4) (2017) 2361–2374, http://dx.doi.org/10.1109/TNET.2017.2686839.

[4] A. Ahmed, L.M. Boulahia, D. Gaiti, Enabling vertical handover decisions in heterogeneous wireless networks: a state-of-the-art and a classification, IEEE Commun. Surv. Tutorials 16 (2) (2014) 776–811.

[5] F. Shi, K. Li, Y. Shen, Seamless handoff scheme in wi-fi and wimax heterogeneous networks, Future Generation Comput. Syst. 26 (8) (2010) 1403–1408.

[6] P.H. Ho, Y. Wang, F. Hou, S. Shen, A study on vertical handoff for integrated wlan and wwan with micro-mobility prediction, 2006 3rd International Conference on Broadband Communications, Networks and Systems, (2006), pp. 1–11.

[7] T. Taleb, A. Ksentini, Vecos: a vehicular connection steering protocol, IEEE Trans. Veh. Technol. 64 (3) (2015) 1171–1187, http://dx.doi.org/10.1109/TVT.2014.2327241.

[8] Y. Kim, H. Ko, S. Pack, W. Lee, X. Shen, Mobility-aware call admission control algorithm with handoff queue in mobile hotspots, IEEE Trans. Veh. Technol. 62 (8) (2013) 3903–3912.

[9] S. Céspedes, X. Shen, On achieving seamless ip communications in heterogeneous vehicular networks, IEEE Trans. Intell. Transp. Syst. 16 (6) (2015) 3223–3237.

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

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

[12] P.M. Santos, T.E. Abrudan, A. Aguiar, J. Barros, Impact of position errors on path loss model estimation for device-to-device channels, IEEE Trans. Wireless Commun. 13 (5) (2014) 2353–2361.

[13] R.M.B. Reza Zekavat, Handbook of Position Location:Theory, Practice and Advances, First, O'Reilly, 2012.

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

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

[16] R. Parker, S. Valaee, Vehicular node localization using received-signal-strength indicator, IEEE Trans. Veh. Technol. 56 (6) (2007) 3371–3380.

[17] A.P. Subramanian, P. Deshpande, J. Gao, S.R. Das, Drive-by localization of roadside wifi networks, INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, (2008).

[18] 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).

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

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

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

[22] A.R.R. Franco, WiFi Roaming Along Urban Routes, Master's thesis, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal, 2014.