Occupancy Estimation using WiFi: A Case Study for Counting Passengers on Busses

Ubaid Mehmood*, I. Moser*, Prem Prakash Jayaraman* and Abhik Banerjee*

* Department of Computer Science and Software Engineering
Swinburne University of Technology
Melbourne, Australia
Email: (umehmood, imoser, pjjayaraman, abanerjee)@swin.edu.au

Abstract—Occupancy estimation in public busses is growing in importance due to its usefulness in planning public transport systems effectively. Bus occupancy data can be used in real time by transport companies to better estimate demand, improve customer satisfaction while also enabling long-term analyses for the purpose of route optimisation. Additionally, in scenarios where busses are used to replace other modes of public transport, such as metropolitan trains during schedule/unscheduled stoppages, occupancy estimation can help operators determine the deployment requirements of replacement busses. In this paper, we present a low cost WiFi-based system for occupancy estimation in busses. In particular, we propose a novel algorithm for occupancy estimation that reduces the problem of overestimation encountered in WiFi-based occupancy estimation approaches. We validate the accuracy and efficiency of the proposed algorithm and the WiFi-based occupancy estimation solution via a real-world experiment.

Index Terms—WiFi, Occupancy Estimation, Passenger Counting, Mobile Sensing

I. INTRODUCTION

Bus operators are interested in the occupancy levels (i.e., number of passengers on a bus) of their bus services to be able to better manage a service - increase or decrease the number of busses or change the size of a vehicle to avoid overcatering, which increases cost, and undercatering, which affects customer satisfaction.

In most situations, public transport requires passengers to pay a fare when entering the vehicle. In developed nations, more often than not the payment is made electronically, for example using an electronic ticket (such as myki in Melbourne, Australia, EZ-Link in Singapore). This provides an instantaneous and mostly accurate count of passengers riding the bus. However, there are cases where such payment systems are not used. Replacement bus services are provided when another service, such as rail, is not available during breakdowns (unscheduled) or scheduled maintenance. Replacement services are often crowded and unpopular, hence transport authorities avoid demanding payment from customers. Similarly, school busses and other chartered services do not require per-passenger payment. In such cases, operators rely on manual counting to understand the usage patterns and demands for the replacement bus services.

88% of Australians owned a mobile smartphone in 2017, while Ireland, Norway, Luxembourg and the Netherlands were reported as having above 90% penetration rates, which are set to rise [1]. Therefore, a WiFi-based system for counting passengers in busses using their mobile smartphones has the potential to deliver very high accuracy (> 90%). This is supported by a number of commercial products for people counting in scenarios such as shopping malls, stadiums, which all report around 95% accuracy [2]–[5]. There are a number of research articles on WiFi sensing for determining the exact location of people [6], [7] as well as determining the number of visitors to an event or shop [8], [9]. The closest related study was conducted by Mikkelsen et al. [10], who used the delays between the first and last times a device was detected and the RSSI (signal strength) values to determine whether a device was on a bus. At times this study overestimates the number of occupants by a factor of 1.5 and higher.

One of the main challenges in estimating passenger numbers in a bus is WiFi devices carried next to the bus and in neighbouring vehicles often leads to overestimation. To address this challenge in this paper, we propose, develop and validate a WiFi-based occupancy estimation system. The system we propose uses data from several low-cost WiFi sensors deployed on the bus (as depicted in Fig. 1) to estimate the number of people in the bus. Our novel system incorporates a smart occupancy estimation algorithm that fuses data from the WiFi sensors to mitigate the overestimation problem. We conducted a real-world trial of the system to validate the accuracy of the proposed occupancy estimation algorithm.

II. BACKGROUND

Among the people counting studies on public transport, Myrvoll et al. [11] investigated the use of WiFi for detecting devices on a bus in Norway. The approach focused on counting people on long-distance coaches and relied on a passenger remaining on a coach across several stops before counting them among the occupants. Another study [12] used WiFi to count people at a bus stop and aboard a bus with the goal of identifying wait times. Mobiles that were sensed only once were excluded from the results. Brandon [13] also attempted
to estimate the number of passengers on a bus, but devised a crowd sensing app and achieved 88% accuracy using a binary classification approach.

Mikkelsen et al.’s study [10] was based on a study by Handte et al. [14], who decided whether a device was on the bus based on the time between the probes. The researchers decided to discard a device if it was not detected twice in 3 minutes. This led to a 20% accuracy, which prompted Mikkelsen et al. to include RSSI values in their algorithm.

Yoshida and Taniguchi [8] estimated the number of people in a laboratory by reading the RSSIs of user devices connected to an access point and analysed the data with a support vector machine (SVM), attaining a 77% accuracy. A similar study by Depatla et al. [9] includes a mathematical model of signal loss and reports 96% accuracy indoors and 63% outdoors. RSSIs have also been used to estimate the progress of queues [15]. WiFi and Bluetooth data was used to infer the number of people at an airport [16] with reported average accuracy of 75%.

WiFi fingerprinting uses several APs to capture messages from a device. Based on the RSSI values, the location of the device is estimated. Youssef and Agrawala [6] devised a fingerprinting system that creates a map of the area of interest and uses a clustering algorithm on distance data provided by several scanners. Test areas of 68.2 * 25.9m inside a university with corridors and rooms on several levels were designated. Using 21 Cisco APs, they report accuracies in the estimated locations between 0.6 and 1.3 metres. Kjærgaard et al. [7] investigated communities of persons using signal strength features and a clustering algorithm to determine whether a person belonged to a group. The accuracies they state pertain to the correct identification of groups, but the authors also report evidence of the number of AP devices having an influence on the accuracy.

While several studies as described in this section have used data sensed from WiFi for people counting, such studies have not solved the overestimation problem. Furthermore, approaches that use WiFi fingerprinting require constant training/re-training to deliver accurate results. On the contrary, our proposed solution uses several low-cost WiFi sensors and incorporates a smart occupancy estimation algorithm significantly reducing the effect of overestimation. Moreover, our approach requires no training/re-training hence, allowing it to be deployed in any kind of bus/coaches.

III. WiFi-based Occupancy Estimation Algorithm

WiFi-enabled devices send probe requests at 1 - 3 minute intervals, depending on whether the device is being used. The probes include the MAC address of the device’s network card, a timestamp and the RSSI value (signal strength). The RSSI values are highly variable and differs for each WiFi-enabled device.

The solution we propose fuses data from several WiFi sensors deployed on the bus to continuously scan for WiFi probe packets. Based on the received probe packets, the algorithm (as presented in Algorithm 1) defines a WiFi-enabled devices as on the bus if:

- all WiFi sensors detect its probes in the same time interval;
- all WiFi sensors receive an RSSI greater than given threshold $l_{rssi}$;
- the earliest and latest probe are a defined number of seconds apart; and
- it is detected at least $l$ times.

The $q$ WiFi sensors collect probe requests over a period of $s$ seconds. The probes captured over an interval $x$ are denoted as $P_{xj} = \{p_{xj1}, p_{xj2},...\}$ for the $j$th WiFi sensor, their MAC addresses as $p_{MAC}^{xj}$, the timestamps as $p_{t}^{xj}$ and the RSSI values as $p_{rssi}^{xj}$.

Algorithm 1 analyses the probe requests detected by each WiFi sensor. The $q$ WiFi sensors provide a new set of probes $P_{xj}$ every $s$ seconds. Probes that have an RSSI value above a threshold $l_{rssi}$ are added to the set $M_{xj}$ of MAC addresses for this WiFi sensor $j$ and interval $x$. The intersection of this set and the set common to all WiFi sensors for this interval $M_x$ is taken. Lines 16 and 21 store the probes with the earliest and latest timestamps for the same MAC address. The method $findProbeWithMatchingMac()$ finds the probe with the same MAC address in the set for the earliest probes $P(e)$ and replaces the timestamp if it is later than the current probe’s. It then finds the probe with the matching MAC address in the latest timestamps $P(l)$ and replaces it if it has an earlier timestamp. If no probe exists in these sets, the current probe is stored.

When line 24 is reached, all addresses observed in interval $x$ by all WiFi sensors with good signal strength are contained in set $M_x$. Lines 24 to 29 add these addresses to the main result list $M$ for this section of the bus trip between two stops. If a MAC address does not exist in set $M$, it is added (line 28), along with a counter (line 29). If a MAC address already exists in $M$, the corresponding counter in the list of counters $C$ is incremented (line 26).

When the bus arrives at the next stop, no further WiFi sensor data probes are processed. MAC addresses are removed from the final list $M$ if they have not been sensed a predefined number of times $l$ or the first and last relevant probes are closer in time than the predefined number of seconds $l_s$ (lines 30 – 36).

![Fig. 1. Illustration of 4 WiFi sensors deployed on the bus](image-url)
Algorithm 1: Occupancy Estimation Algorithm

Data: \( l, l_{tssi}, l_s \)

1. \( M \leftarrow \emptyset \) // List of on-bus devices
2. \( C \leftarrow \emptyset \) // List of counters
3. \( M_j \leftarrow \emptyset \) // List of MAC common in interval
4. \( P(e) \leftarrow \emptyset \) // Probes with earliest timestamps
5. \( P(l) \leftarrow \emptyset \) // Probes with latest timestamps
6. \( x \leftarrow 0 \)

Result: \( M \) // MAC of devices defined as on bus
begin

while between bus stops do

\( x \leftarrow x + 1 \) // counts intervals of \( s \) seconds

\( M_j \leftarrow \emptyset \)

/* Find probes common to \( q \) sensors */

for \( j \leftarrow 1 \) to \( q \) do

foreach \( p_{xji} \in P_j \) do

if \( p_{xji}^{x} \succ l_{tssi} \) then

\( M_{xj} \leftarrow M_{xj} \cup \{p_{xji}^{MAC}\} \)

\( y(e) \leftarrow \) findProbeWithMatchingMac\((P(e))\)

\( y(l) \leftarrow \) findProbeWithMatchingMac\((P(l))\)

if \( p_{xji}^{y} < y(e) \) then

\( y(e) \leftarrow p_{xji} \)

if \( p_{xji}^{y} > y(l) \) then

\( y(l) \leftarrow p_{xji} \)

\( M_j \leftarrow M_j \cap M_{xj} \)

/* all probes of interval list */

foreach \( m_{xj} \in M_j \) do

/* Go through result list \( M \) */

for \( k \leftarrow 1 \) to \( n \) do

// Check if MAC in result

if \( m_{xj} = m_{xk} \) then

\( c_k \leftarrow c_k + 1 \) // increment counter

else

\( M \leftarrow M + \{m_{xj}\} \)

\( C \leftarrow C + 1 \) // add new counter

/* Go through result list \( M \) */

for \( k \leftarrow 1 \) to \( n \) do

if \( c_k < l \) // check occurrences

\( M \leftarrow M \setminus \{m_k\} \)

\( C \leftarrow C \setminus \{c_k\} \)

\( y(e) \leftarrow \) findProbeWithMatchingMac\((P(e))\)

\( y(l) \leftarrow \) findProbeWithMatchingMac\((P(l))\)

if \( (y(l) - y(e)) < l_s \) // check time difference

\( M \leftarrow M \setminus \{m_k\} \)

end

IV. IMPLEMENTATION AND EMPIRICAL EVALUATION

A. Implementation

The WiFi sensors used in this trial were built from off-the-shelf components that included:

- ESP8266, available for AUD $10,
- a battery pack available for AUD $20 and
- a SD card reader available for AUD $5 that stores the probes received.

Figure 2 presents an illustration of the deployed WiFi sensor with a battery pack in a bus.

B. Trial Setup

Four WiFi sensors \( (q = 4) \) were configured in monitor mode and placed in the bus as shown in Fig. 1, in the front \((F)\), back \((B)\) and left and right sides \((L)\) and \((R)\). For the purpose of evaluating the accuracy of the occupancy estimation algorithm (Algorithm 1) proposed, we evaluated all possible combinations of WiFi sensors. In the ensuing discussion, each configuration is referred to by the first letter of the WiFi sensor location in the bus. For example, RLB refers to the case where only probe packets from the three WiFi sensors deployed at the right, left and back of the bus were considered by Algorithm 1 to estimate the number of passengers in the bus. Fig. 3 shows the WiFi sensor setup in the bus used for experimental trials.

The threshold that defines the minimum number of probes that has to be received by all WiFi sensors was set to \( l = \)
where \( l_s \) was set to 160. The minimum RSSI recorded by each WiFi sensor for a device’s probes was set to \( l_{r_{\text{RSSI}}} = -99 \).

In order to evaluate the algorithm, we conducted multiple trials on a bus in a real-world setting. The trial included participants comprising of volunteers including university staff and students. The bus made multiple trips in a rectangular route measuring approximately 8 km around the Hawthorn campus of Swinburne University of Technology. The route was chosen in a manner that it included segments of low, moderate and heavy traffic. This included stops of approximately 10 minutes after each trip. The trip route is depicted in Fig. 4. Occupancy estimation was performed for each trip. A total of 6 trips were undertaken, with the occupant count varying between 10 and 17. Passengers were counted manually to establish the ground truth.

### C. Performance Metric

We measured the occupancy estimation performance in terms of the ratio of the estimated count to the actual count, which we termed \( \text{Occupancy Estimation Ratio}, E_R \), expressed as below:

\[
E_R = \frac{|M|}{C_g}
\]

(1)

where \( |M| \) is the count of people estimated using Algorithm 1 and \( C \) is the ground truth of the number of people in the bus. Having \( E_R > 1.0 \) implies overestimation of the bus occupancy and as observed in literature [10], this can be quite high at times owing to signals from people not in the bus as well as other vehicles on the road. While underestimation, i.e. \( E_R < 1.0 \), of the occupancy count is likely to be equally misleading, we focus on addressing the problem of overestimation primarily because of two reasons. Firstly, overestimation is a characteristic of occupancy estimation in vehicular scenarios. Secondly, underestimation of the occupancy count may be addressed in combination with other techniques and also includes factors such as people not carrying a mobile device. For instance, participatory techniques in which commuters volunteer to provide occupancy information could be used in conjunction with WiFi based estimation. However, such data would not be useful in removing information about other vehicles that contribute to overestimation.

### D. Evaluation Results

We have evaluated the performance in terms of \( E_R \) for all possible combinations of WiFi sensors.

In Fig. 5, we compare the mean value of \( E_R \) for each combination computed across all trips. Using a single WiFi sensor, irrespective of its placement, results in an overestimation by a factor greater than 10 on average. Looking at the results for each trip in Fig. 6, it is apparent that using a single WiFi sensor results in overestimation of at least 5 times the ground truth. However, combining the observations from multiple WiFi sensors leads to a significant reduction in overestimation.

Having established that observations from multiple WiFi sensors are necessary for mitigating the impact of devices outside the bus leading to overestimation, we now investigate how many WiFi sensors are required to maximise accuracy. While increasing the number of WiFi sensors to 2 reduces the mean value of \( E_R \) as compared to having single WiFi sensors, the ratio is still well over 2.5. Given this, we take a closer look the using 3 and 4 WiFi sensors for occupancy estimation. Using 3 WiFi sensors brings down the mean \( E_R \) to below 2, though there are still instances with high values of \( E_R \), going as much as 7.69 in one case. However, the configuration RLB is observed to have a higher consistency that other 3 node configurations, with \( E_R \) ranging between 0.77 and 1.31, which is interesting, because the right and left WiFi sensors are close to each other and are not expected to differ much in their data collection. Using 4 nodes results in consistent performance across all trips, with only one instance of overestimation with a ratio of \( E_R = 1.15 \) in Trip 4.
V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a low cost WiFi-based system for counting passengers in busses. The system we proposed incorporates a novel occupation estimation algorithm. Experimental evaluations in a real-world setting show that the overestimation problem due to interference from external devices can be mitigated considerably by combining probe request data from several WiFi sensors. In particular, we find that using four anchors together results in a maximum overestimation of 15% (i.e. \( ER = 1.15 \)). Using three anchors also brings down the impact of overestimation, although to a lesser extent, with the mean overestimation across all three-anchor configurations obtained at 67% (i.e. average \( ER = 1.67 \) across all three-anchor configurations).

The WiFi sensors are inexpensive and can be combined with a microcontroller and hotspot for real-time processing and reporting, rendering SD cards unnecessary.

In future work, we will devise such a system. We are currently investigating a scenario where busses transport members of the public on actual bus routes to test the approach with authentic peak and off-peak scenarios. The number and placement of WiFi sensors are varied in these trials to determine the optimal configuration of WiFi sensors in a bus.

ACKNOWLEDGEMENT

This research is funded by iMOVE CRC, Transport for NSW and Sydney Trains and supported by the Cooperative Research Centres program, an Australian Government initiative.

REFERENCES

