Public Transport Occupancy Estimation Using WLAN Probing

Mikkelsen, Lars Møller; Buchakchiev, Radoslav Naskov; Madsen, Tatiana Kozlova; Schwefel, Hans-Peter

Published in:
Resilient Networks Design and Modeling (RNDM), 2016 8th International Workshop on

DOI (link to publication from Publisher):
10.1109/RNDM.2016.7608302

Publication date:
2016

Document Version
Peer reviewed version

Public Transport Occupancy Estimation Using WLAN Probing

Lars Mikkelsen* lmm@es.aau.dk, Radoslav Buchakchiev* rbucha13@student.aau.dk,
Tatiana Madsen* tatiana@es.aau.dk and Hans Peter Schwefel* hps@es.aau.dk
*Wireless Communication Networks
Aalborg University, Denmark

***Abstract*—Prediction of availability of physical services can be a valuable addition to transportation systems operation. In this paper we are focusing on estimation of public transport occupancy (PTO), or more specifically, on estimating bus passenger load, i.e., the number of people on the bus. This information can be used by bus operators as input to the analysis of bus routes' efficiency, or to provide an app indicating passenger load. PTO estimation based on collecting WiFi probes emitted by WiFi enabled devices is cheap and easy to install. This paper presents a prototype implementation of this method, analysis of the collected data and of the estimation algorithm accuracy. Analysis of passenger load in a bus has indicated that there are two main challenges of the estimation using WiFi probes. The algorithm provides overestimation due to inclusion of WiFi devices that are outside the bus and underestimation due to exclusion of people without an active WiFi enabled device or by missing out probes in the detection algorithm from devices carried on board. We have shown how by fine-tuning parameters of the algorithm the probes received from people outside the bus can be filtered out thereby reducing the severity of the underestimation problem. The typical approach to combat the overestimation problem is to make the adjustments based on a statistical ratio of people possessing a WiFi enabled smart device over the whole population.**

***Index Terms*—WiFI probes; bus occupancy; RSSI**

I. Introduction

Optimization of transportation services and passenger information services can benefit from real-time information about vehicle occupancies. The latter can be obtained using visual inspection, checkpoint counting (e.g. via special ticketing systems), or inferred from communication systems. Visual inspection means looking at an area and e.g. counting the number of people. This can be problematic due to legal restrictions on cameras in the public space. The checkpoint counting is limited to physical checkpoints, such as entrances to metro stations or other, where in the more free space areas it has limitations. Communication system inference means extracting information about number of users connected or using different communication systems, and from this give an estimate of actual number of people. Such occupancy estimators depend on the penetration rate of the communication systems, i.e. the fraction of the number of people that carry a device that has such technology activated.

Despite the obvious inaccuracies of the last approach it has a clear advantage; cost. By using an already widely implemented and used system, the cost of realizing people density estimating can be greatly reduced.

In this work we will give an estimate of the number of people by collecting WiFi probes that WiFi enabled devices emit to discover WLAN APs. We will investigate the feasibility of the solution when sensors are placed on a number of buses driving in a town. This will enable estimation of the number of people on the bus, and the number of people in the immediate vicinity of the bus giving the necessary input to two different services: 1) providing info on bus load to public transport users for trip planning purposes; 2) providing updated city wide image of the number of people in the different parts of the city.

As smart devices equipped with Bluetooth and WLAN communication technologies are becoming more and more widespread, the research focus is directed on the opportunities for utilizing signaling messages of such communication technologies for estimation of number of people. Similar approaches have been applied for cellular technologies in [1], [2], [3]. The way how a mobile device searches for nearby APs, with WiFi probes, or how beacons related to Bluetooth device discovery procedure are transmitted (explored in [4]), can be exploited to estimate the number of active devices in a certain area.

These approaches allow discovery and counting people carrying a WiFi or Bluetooth capable device, , when the device has the communication technology activated, and people without such devices or with WiFi or Bluetooth turned off are not counted. However, the number of smart devices is constantly growing, e.g. Cisco [5] predicts that 53% of the IP traffic in 2019 will be generated from WiFi devices, as well as the number of WiFi hotspots will grow each year from 64.2 million in 2015 to 435.2 million in 2020. These trends make approaches utilizing WiFi communication technologies realistic and promising for people density estimation.

Bluetooth device discovery procedure is based on the process when Bluetooth devices send discovery requests periodically to find the devices nearby, and if a device is not in a "hidden" mode, it will send discovery reply. Recently, in the literature it has been reported several implementation of systems based on counting of Bluetooth devices. In [6] a scanner device is developed and installed in buses to capture the discovery requests and it subsequently uses MAC address as a unique identifier. The data is used to create origin/desti-

nation matrix to improve bus utilization. A similar approach is considered in [7] for counting the attenders of a European soccer championship.

Currently WiFi communication interface is even more used compared with Buetooth interface, making it easier to detect the presence of a smart device via WiFi probes. [8] estimates crowd density and people flow at a major German airport. Scanner devices are put before and after a security check and listen for probe requests. The number of boarding passes on security check is used as a ground truth which allows estimation of developed algorithms accuracy and fine-tuning algorithm parameters. In [9] a use case of people density estimation in a public bus system is considered. It argues that providing information to the passenger, including trip info and the expected number of people on a bus can enable the travelers to optimize their comfort. [9] reports the results collected by the deployed system in three buses in the city of Madrid during 3 weeks. The focus of the analysis is on the possibility of obtaining data near real time and the accuracy of the estimation algorithm. It is however difficult to provide algorithm accuracy as it requires knowing the ground truth of the number of devices with activated WiFi interface in a bus and it is very difficult to obtain this information.

In this paper we consider the scenario similar to [9]. We design a low-complexity threshold based estimator and focus on the estimation algorithm accuracy. We perform manual counting of passengers in a bus during system trials and use it to compare with our estimations. This allows to perform the analysis what impact different parameters in the algorithm have on its accuracy and how to fine-tune them in order to achieve acceptable accuracy.

It is worth mentioning that there exist a number of commercial implementations of user tracking and counting using WiFi probe requests, e.g. Cisco Meraki CMX(Connected Mobile Experiences) Location Analytics [10] and Blip Systems [11]. Cisco Meraki uses the data to track people for successful marketing campaigns, while Blip Systems is focused on counting people in a queue or road traffic monitoring.

A. Outline

The rest of the paper is organized as follows. Section 2 provides the details of the system design and outlines the simple algorithm used for bus passenger load estimation. Results of the measurement campaign are presented in Section 3, together with the detailed analysis of the parameter impact of the algorithm performance. Section 4 concludes the paper with discussions and outlook for future work.

II. System design

In this section we will describe how the system for collecting WiFi probes is designed, along with how we anonymize the collected data, how we collect the data, and how we process it to obtain people density estimation.

The system, depicted in Figure 1, contains 3 main modules, divided according to responsibilities; sensor node, collector server, and processing service. A sensor node is placed on a bus, and is collecting WiFi probe requests emitted by WiFi enabled devices in the vicinity of the sensor. Probe requests contain the identifiable information for a device (a MAC address). Subsequently, collected identifiable information of the probe is anonymized, and sent to the Collector Server. The Collector Server receives and stores the collected probes, and makes them available through a interface to the Processing Service. The Processing Service fetches the collected probes via the interface and processes them. The goal of the processing is to filter probes from devices that are on the bus.

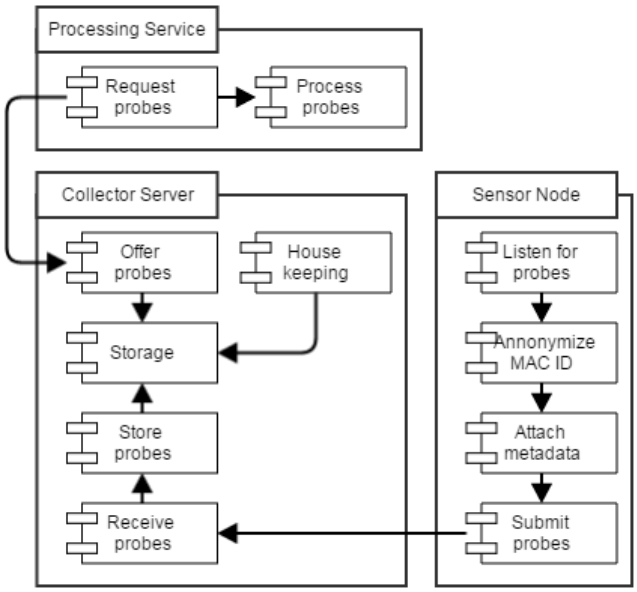

Fig. 1. WiFi probe catching system overview and interactions.

This design and division of functionalities provides us with great flexibility. By having a central collector node it will operate as a gateway for the captured probes, allowing us to scale the number of sensor nodes easily. Furthermore, if the probes should be processed in another way we simply develop another service to do this. This is particularly usable in the IoT setting where information from one domain can be re-purposed in many other domains. The collector node is a single point of failure, but can easily be replicated with multiple instances submitting probes to the same database.

1) Sensor Node: The Sensor Node has the following modules; scanner, anonymization, storage, submit to collector. The scanner is always on, listening for WiFi probes from devices. When a probe request is received it is sent to the anonymizer, which applies a hashing algorithm using the date, a secret key, and the MAC of the probe request to obtain a scrambled ID. From the scrambled ID only a subsequence is used, which ensures that it is non-reversible. It does however mean that there is a chance for collisions, meaning that different MAC addresses could return the same scrambled ID. This does seem like a fair trade off for ensuring privacy of users. After the anonymization the probe with the new scrambled ID is stored

in a SQLite storage until it is submitted to the Collector Server. The SQLite is chosen for its low resource usage and ease of implementation. When submitting to the Collector Server we also attach a Sensor ID such that probes captured from different sensors can be processed separately.

Each probe request is combined with some additional information from the sensor, such that each probe request will contain the following information:

- MAC address of the device (scrambled after anonymization)
- RSSI
- GPS location of the sensor
- time stamp from when received at the sensor
- sensor ID

The implementation of the Sensor Node is done using a Raspberry Pi 1 model B as the main device. To this we connect a GPS receiver for location services, a WiFi dongle for scanning of probes, and a GSM shield for connectivity. The Raspberry Pi runs Raspberian and the software is written in Python.

2) Collector Server: The Collector Server has the following modules; receiver, storage, cleanup, server. The receiver is the interface receiving input from the sensor nodes. This will be realized as a RESTful interface, allowing sensor nodes to POST JSON formatted probe request data. Furthermore, to ensure data integrity the interface will be protected via authentication and message integrity protection. After receiving probe request data it is saved in the storage.

The cleanup module will on a periodic basis clean old probe request data. This is to comply with the regulation that probe request information cannot be stored for more than 24 hours, but also to keep the storage requirements low [12] .

The server module makes the collected probe request data available to other services. This is allowing for request of probe requests based on sensor ID, location or area, time interval, or a combination of these.

The Collector Server is realized using Apache CouchDB [13] as this supports all the necessary functionalities needed from the design.

3) Processing Service: The application possibilities of the probe request are many, but in our case we use it to try to estimate the number of people on the bus. For this we need the probe requests captured from one sensor node. This data will both contain probe requests from devices on the bus, but also from devices outside the bus. Therefore we need to apply a filtering algorithm that estimates which devices are on the bus, and when. We do this by using the scrambled ID, the time stamp, and the RSSI.

The design principles behind the estimation algorithm are the following:

- a bus passenger stays on the bus for some time and the probes from his/ her device will be detectable for some time
- Probes from devices on the bus will have higher RSSI than probes from devices outside the bus

One should note that the WiFi standard does not define when or how often probe requests have to be sent by a device. The frequency of the probes varies depending on the network card, driver, mobile device, operating system and application used. According to Cisco CMX Analytics [10], probe request interval for smartphones is around once a minute if a device is in sleep mode (screen off) and 10-15 times a minute for standby mode (screen on). Laboratory tests [14] done for several brand of mobile phones with different operating systems support these results and indicated large variability in probe frequency. For this reason, we do not introduce any assumptions on how often we expect to receive probes from a device.

Based on these assumptions we now define a simple algorithm to perform the filtering of the probe requests. The input to the algorithm will be probes captured from one sensor node in a limited time interval. Furthermore, the probes are prefiltered by only using probes that have a RSSI higher than some threshold. The output is a list of start and end times of devices traveling with the bus. If the time interval between the first and last times the device is detected is longer than the minimum duration interval, then the device is counted as being on a bus during that period. The estimation algorithm is summarized in the following:

```
list all device ids
for each device id
 list probes with RSSI > B_{threshold}
 for each probe
  set as initial
  find time to all following probes
  if time to a probe > T_{threshold}
   set device id as in bus
   break
```

III. Results

In this section we will present some collected probes and evaluate the output of the occupancy estimation algorithm.

A. About captured probes

To test the system we made 4 test runs on a city bus line (2 different days and 2 each direction) with a sensor node. In the Sensor Node the GPS receiver is a u-blox 7 UBX-G6020, and the WiFi dongle is a TP-LINK TL WN722N. The GSM module is not implemented, since we can collect the probes locally during the test-run. For the same reason we have not realized the Collector Server for the test run.

While the sensor captured probe requests we were manually counting the number of people on the bus. This manual count will be compared to the filtering result, and will be referred to as Ground Truth. The collected probes from these 4 test runs are summarized in Table I and in Figures 2 to 5.

In the 4 test runs we collected probes with between 575 and 799 unique devices IDs. 94% of devices transmit at most 20 probes, and 59% of devices transmit more than 1 and at most 20 probes.

TABLE I
COLLECTED PROBES PER TEST RUN (*=DETAILED ANALYSIS).

Route	From AAU	From City
27/4	5302*	5295
10/5	5785	5364*

We would like to note that the bus route has been selected that goes from the university campus to the city center and beyond, passing first through campus and suburb areas, then arriving to the city center and train station and continuing through the city district with apartment buildings. Thus, this route is very inhomogeneous in terms of people presence on the streets and houses/apartments density along the road.

Due to the lack of space we have decided to present a detailed analysis of 2 sets of data: Test 1 done on 27/4 bus route starts from University (AAU) and Test 2 done on 10/5 from the city. Figure 2 shows the amount of all detected devices (except those seen only once) compared to ground truth value that was obtained by manually counting people. We have chosen these two cases to concentrate our analysis on as they show different type of behavior. For Test 1 (see Figure 2 top) the number of detected devices lies around 20 devices with some variations, while the real number of people in the bus was constantly increasing and over 30 people were going until the last bus stop. For Test 2 (see Figure 2 bottom) the bus was not crowded, with maximum of 30 people and the people were leaving the bus gradually as it was passing the suburban areas. One can observe a large peak in the number of detected devices in the middle of the route. This fits very well with the time when the bus was passing through the center with the train station; a lot of bus stops and cafes on the streets. The fact that we do not see this spike on the previous graph can be explained that the tests were done during different time of a day.

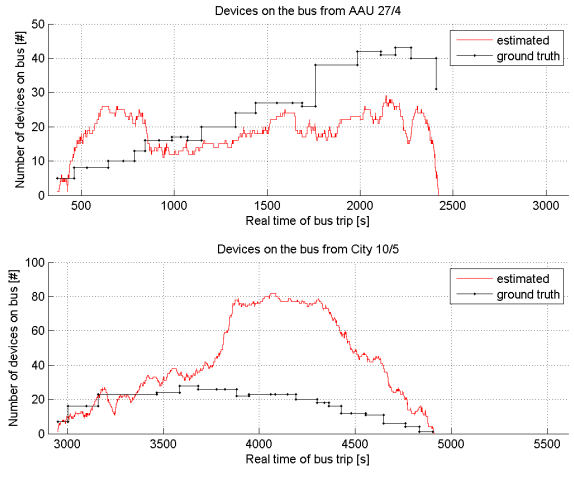

Fig. 2. Presence of all detected devices aggregated as initial estimated number of people.

B. Impact of the active time threshold/ Impact of the minimum visibility duration interval

First, we investigate what impact has the threshold value of the active time interval on the algorithm accuracy. From Figure 3 we can see plots of the duration of the visibility of devices, i.e. the time between first and last probe pr ID. Note that these graphs include only devices that are seen at least twice and the time interval between the first and last seen probes is longer than 30 sec. The threshold of 30 sec is selected to ensure readability of the graphs; otherwise the graphs are becoming overcrowded. This can be seen from Figures 4 and 5 that presents histograms of the lifetime of all the probes, meaning the time from the first probe to the last per ID. Note that the devices that are "seen" only once are not included on the graphs. From these figures one can see that approx. 80% of detected probes have a lifetime less than 30 sec. This indicates that a lot of devices outside a bus are detected (or that people carry multiple WiFi enabled devices) and these devices should be filtered out.

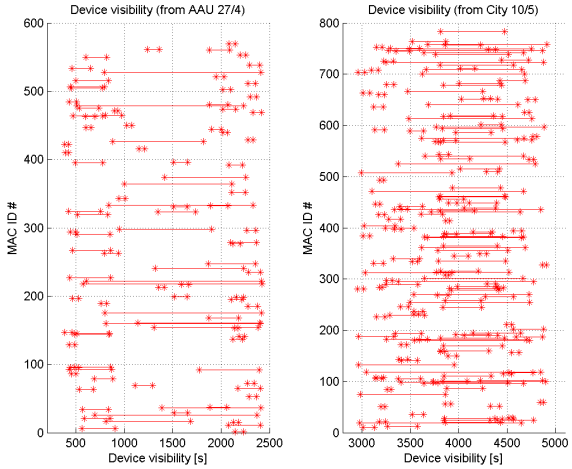

Fig. 3. Device visibility duration (minimum time 30s).

The simple way for filtering out unwanted signals is to consider a minimum visibility duration interval. In Figure 6 we evaluate the impact of setting different threshold values on this parameter.

From these plots we can see that the filtering has only a limited impact on the accuracy. Filtered graphs does not follow well the Ground truth graph, both in terms of tendencies and distance between graphs. Setting the parameter to 30 or 60 sec helps to remove small variations caused by detecting almost static objects such as pedestrians. There is a tendency that the longer the minimum time the more of the extreme peaks are filtered away. This makes sense if the extreme peaks are due to specific areas in the city with higher density of people. However, we observe that some areas with large overestimation are not removed. This overestimation could be due to another bus driving behind, or slow bus speed in the city center and longer time intervals spent on the bus stops.

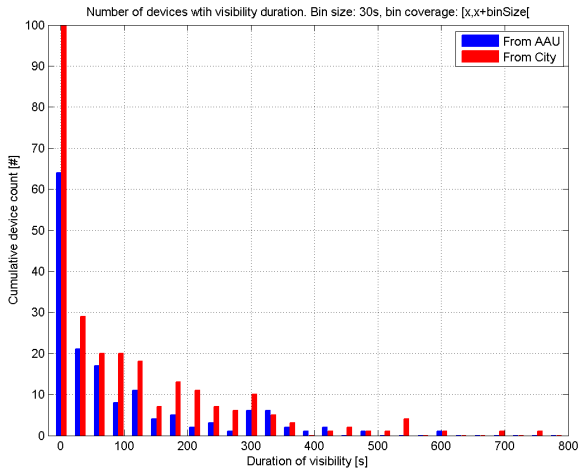

Fig. 4. Histogram of device visibility duration for 27/4.

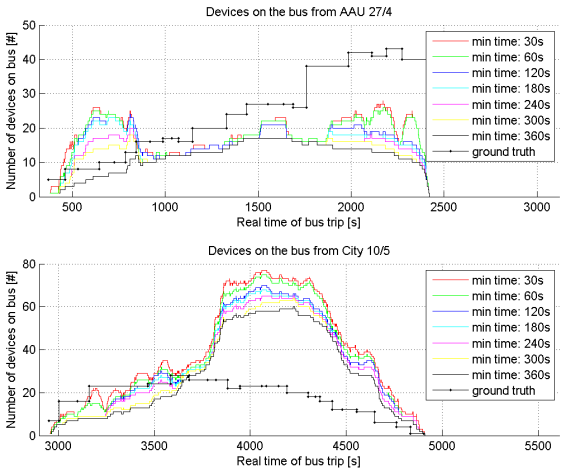

Fig. 6. Number of devices filtered with different minimum times.

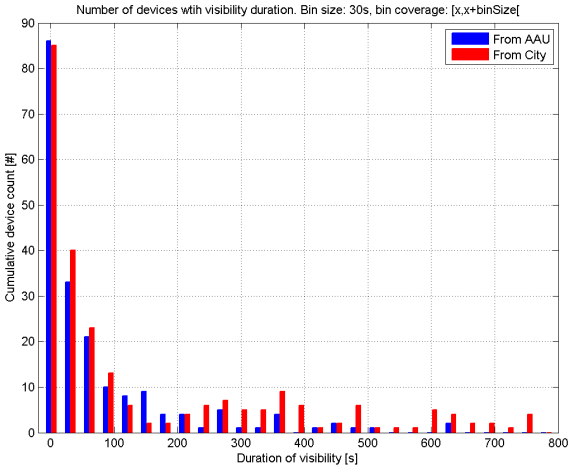

Fig. 5. Histogram of device visibility duration for 10/5.

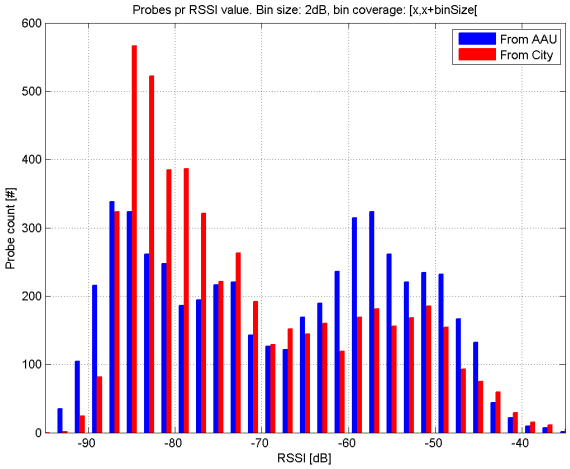

Fig. 7. RSSI Histogram for 27/4.

This leads us to a conclusion that using only visibility duration interval is not enough.

C. Impact of RSSI threshold

In Figures 7 and 8 we can see a histogram of the RSSI of all the probes. The shape of the histograms has two peaks (around -55 dB and -85 dB) and it leads us to a hypothesis that the RSSI levels for passengers on a bus are distributed around -55 dB and RSSI values for those outside the bus lies around -85 dB. Setting a threshold e.g. at -60 dB will filter many probes away.

In the following we investigate what impact the RSSI value threshold has on the algorithm accuracy. In this approach the initial probe of a device is considered only if it is above some RSSI threshold.

From Figures 9 and 10 we can see that filtering based on RSSI value has great effect. The estimated curves for a passenger load lie under the ground truth which is reasonable as not all people on a bus has a device with an active WiFi. From Figures 9 and 10 we verify how well the estimation follows the ground truth. Assuming that the same percentage of passengers has a WiFi enabled device, the ratio between the estimated number of devices and number of passengers in the bus should be constant. Figure 11 confirm this assumption and shows that for parameter settings of *min time = 6 min* and *min RSSI value = -65 dB* the ratio is around 50%.

IV. Summary and Outlook

Widespread use of WiFi communication technology makes it possible to utilize it to create new exciting services and applications. An example of such a service is a bus load estimation using WiFi probes. This idea is not new and commercial solutions already exists e.g. for people density estimations in airports or traffic jam detections. We are not

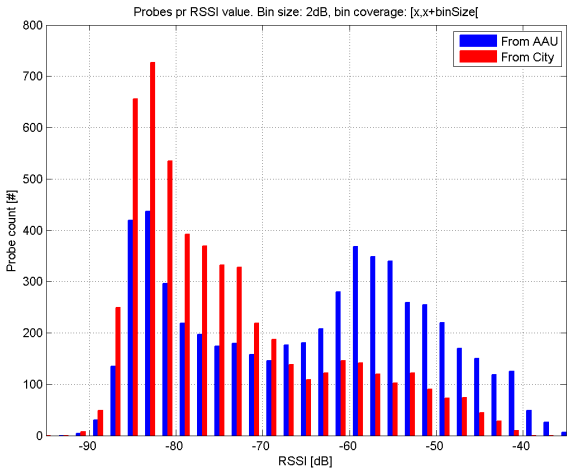

Fig. 8. RSSI Histogram for 10/5.

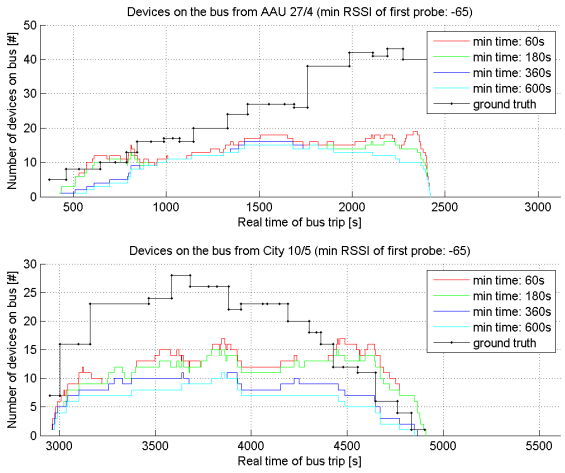

Fig. 10. Number of devices filtered with different minimum times and minimum -65dB.

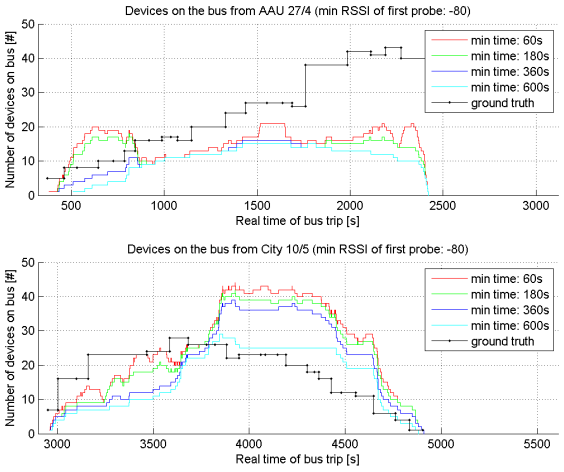

Fig. 9. Number of devices filtered with different minimum times and minimum -80dB.

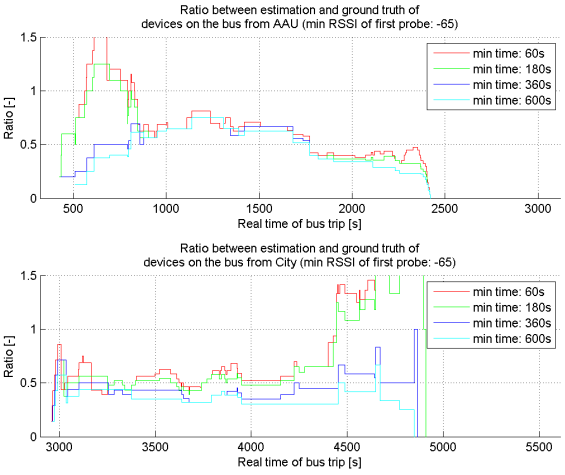

Fig. 11. Ratio between estimated number of devices and ground truth with different minimum times and minimum -65dB for 27/4 and 10/5.

aware of any commercial solution for estimation of the number of people on a public transport unit, only some initial trials, e.g. [9] indicates that such solution is feasible. In this paper we present a low-complexity threshold based estimation algorithm and analyzed its accuracy. our findings indicate that the algorithm performance is very sensitive towards the algorithm parameter settings. In order to achieve high accuracy the parameters related to the minimum values for the time a device is observed and RSSI of the signal from the device, should be fine-tuned. It has been found that $B_{threshold} = -65dB$ and $T_{threshold} = 6min$ gives good estimation, since it helps to filter out unwanted probes from devices outside a bus. However, it might require other parameter settings for other places than Aalborg city with their own specifics of bus routes.

In this paper we have considered a simple algorithm for bus load estimation. Design of more complex approaches e.g. taking into account the location of bus stops and information about route surroundings, utilizing machine learning methods would hopefully give higher estimation accuracy. However there is a trade off between the accuracy and algorithm complexity. Here complexity lies not only in the increase in computational requirements, but also in complexity of obtaining and matching geographical data.

One can consider other methods for people density estimation and apply them for estimation of public transport utilization. This opens up an interesting research question on how to fuse people density information from different sources in this case. The main challenge comes from the fact that different methods would have different accuracy of the estimates requiring introduction of a quality metric. This is a direction for future research.

ACKNOWLEDGMENTS

The work is supported by BIG IoT (Bridging the Interoperability Gap of the Internet of Things) project funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 688038.

REFERENCES

[1] F. Ricciato, A. Janecek, D. Valerio, and H. Hlavacs, "The cellular network as a sensor: From mobile phone data to real-time road traffic monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, October 2015.

[2] F. Calabrese, M. Colonna, P. Lovisolo, D. Parata, and C. Ratti, "Real-time urban monitoring using cell phones: A case study in rome," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 141–151, March 2011.

[3] N. Caceres, J. P. Wideberg, and F. G. Benitez, "Deriving origin-destination data from a mobile phone network," *IIET Intelligent Transport Systems*, no. 1, pp. 15–26, March 2007.

[4] J. Figueiras, H. P. Schwefel, and I. Kovacs, "Accuracy and timing aspects of location information based on signal-strength measurements in bluetooth," in *2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 4, Sept 2005, pp. 2685–2690 Vol. 4.

[5] Cisco, "White paper: Cisco visual networking index: Global mobile data traffic forecast update, 2015-2020," February 2016. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[6] V. Kostakos, T. Camacho, and C. Mantero, "Towards proximity-based passenger sensing on public transport buses," *Springer Personal and Ubiquitous Computing*, vol. 17, no. 8, pp. 1807–1816, December 2013.

[7] J. Weppner and P. Lukowicz, "Bluetooth based collaborative crowd density estimation with mobile phones," in *Proceedings of IEEE International Conference on Pervasive Computing and Communications*, ser. PerCom 2013, 2013, pp. 193–200.

[8] M. Handte, M. U. Iqbal, S. Wagner, W. Apolinarski, P. J. Marron, E. M. M. Navarro, S. Martinez, S. I. Barthelemy, and M. G. Fernandez, "Estimating crowd densities and pedestrian flows using wi-fi and bluetooth." in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services.*, ser. MOBIQUITOUS 2014, 2014, pp. 171–177.

[9] M. Handte, M. U. Iqbal, S. Wagner, W. Apolinarski, P. J. Marron, E. M. M. Navarro, S. Martinez, S. I. Barthelemy, and m. G. Fernandez, "Crowd density estimation for public transport vehicles," in *Workshop Proceedings of the EDBT/ICDT 2014 Joint Conference*, ser. EDBT/ICDT 2014, 2014, pp. 315–322.

[10] Cisco, "Cmx location analytics," 2016. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[11] B. Systems, "Blip track," 2016. [Online]. Available: http://blipsystems.com

[12] A. . D. P. W. Party., "Opinion 13/2011 on geolocation services on smart mobile devices," May 2011. [Online]. Available: http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2011/wp185_en.pdf

[13] Apache, "Couchdb," 2016. [Online]. Available: http://couchdb.apache.org/

[14] J. Freudiger, "How talkative is your mobile device? an experimental study of wi-fi probe requests." in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks.*, ser. WiSec 2015, 2015.