Crowd Density Estimation for Public Transport Vehicles

Marcus Handte, Muhammad Umer Iqbal, Stephan Wagner, Wolfgang Apolinarski, Pedro José Marrón
NES
University of Duisburg-Essen
first.last@uni-due.de

Eva Maria Muñoz Navarro, Santiago Martinez
Investigacion y Desarrollo
ETRA
(emunoz|smartinez).etra-id@grupoetra.com

Sara Izquierdo Barthelemy, Mario González Fernández
Proyectos Europeos
EMT de Madrid
first.last@emtmadrid.es

ABSTRACT

Existing information systems for urban public transportation are empowering travelers to optimize their trips with respect to travel duration. Experience with such systems shows that this is a viable approach. However, we argue that solely relying on trip duration as the primary indicator for satisfaction can be limiting. Especially, in urban settings providing additional information such as the expected number of passengers can be highly beneficial since it enables travelers to further optimize their comfort. As technical basis for determining the number of passengers, we have built an inexpensive hard- and software system to estimate the current number of passengers in a vehicle. Furthermore, we have deployed the system in several buses in the city of Madrid. In this paper, we describe the overall design rationale, the resulting system architecture as well as the underlying algorithms. Furthermore, we provide an initial report on the system's performance. The initial results indicate that the system can indeed provide a reasonable estimate without requiring any manual intervention.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

WLAN Monitoring, Presence Detection, Intelligent Transport Systems, Smart Cities

1. INTRODUCTION

Today, most information systems for urban public transportation are empowering travelers to optimize their trips with respect to travel duration. To do this, they integrate static information about routes and schedules with dynamic information about unexpected delays. On top of this they provide planing engines that compute shortest paths in order to minimize the trip duration for the travelers.

Clearly, past experiences with such systems shows that this is a viable approach that is useful for many travelers. However, we argue that solely relying on trip duration as the primary indicator for traveler satisfaction can be limiting as it hides many other facets that impact the travelers comfort. Examples may include environmental information such as the accessibility of different vehicles for travelers with special needs or dynamic information such as the likelihood of being able to get a seat in a particular vehicle.

Especially, in urban settings where the same destination can be reached over multiple routes or where the same route is traversed by different vehicles frequently, providing additional information can be highly beneficial. For example, considering the former case, a traveler might simply be able to slightly adjust his route whereas in the latter case, a traveler might simply have to start a trip earlier or later in order to improve his or her level of comfort.

Besides from trip duration, a main influential factor for the overall level of satisfaction with a particular public transport option is the overall crowdedness of the vehicles. However, in the absence of a mandatory reservation system or a fine-grained trip-based payment system, capturing the number of passengers is a challenging and costly task that is typically done by means of manual counting. Yet, in order to provide real-time information on a city-scale such manual approaches are clearly ill-suited.

In this paper, we describe an alternative approach to determine the number of passengers in a vehicle. Based on this approach, we have built an inexpensive hard- and software system to estimate the current number of travelers in a vehicle. Furthermore, we have deployed the system in several buses in the city of Madrid. In addition to the estimation of number of travelers, our system also estimates the location of buses between the bus stops. Based on this deployment, we provide an initial report on the system performance. The results indicate that the system can indeed provide a reasonable estimate for the number of passengers inside the vehicle as well a reasonable estimate of the location of buses between two stops.

The remainder is structured as follows. In the next section, we briefly discuss the underlying design rationale. Thereafter, in Section 3, we outline the overall approach. In Section 4, we describe details of our implementation and in Section 5, we report initial results of our deployment in the city of Madrid. In Section 6, we discuss related work and finally,

in Section 7 we conclude the paper with a short summary and an outlook on future work.

2. DESIGN RATIONALE

As described previously, our goal is to provide a system to determine the number of passengers in a particular vehicle of a public transportation system in order to provide the resulting crowd density information to the travelers. As a result of this overarching goal, we can derive the following five sub goals:

- *Sufficient accuracy:* To provide meaningful information, the system should be able to determine the number of passengers accurately. Thereby, it is important to note that given the typical capacity of vehicles the system does not have to be perfect. Instead, smaller deviations can be tolerated as long as the overall tendency of the crowd density reflects the real situation.
- *Full automation:* To be reliable and feasible to deploy, the system should not rely on manual intervention by passengers. Furthermore, it should not put additional stress on the support personnel such as the driver or the guards. Instead, the system should be able to determine the number of passengers automatically.
- *Low cost:* To be scalable to a city level, the hardware cost of the system should be minimal. As a result, the system should only consist of low-cost off-the-shelf components and it should optimally leverage the existing infrastructure.
- *Low latency:* To provide fresh information to the travelers, the system should be able to report changing numbers of passengers quickly such that it can not only be used for advance planing based on historical data but also to support ad hoc decisions by travelers based on the current state.
- *Low privacy impact:* To be acceptable for the passengers of the public transport system, the system should be non-intrusive from a privacy perspective. Furthermore, it should only gather information that is needed to provide the service and ideally, it should be hard to retrofit the information for non-related use cases.

3. APPROACH

Based on the five goals, we describe our overall approach in the following. To do this, we first describe the basic idea and the resulting system architecture. Thereafter, we describe the details of the algorithms used for crowd density estimation and vehicle tracking. In the next section, we describe the implementation details for our deployment in several buses in the city of Madrid.

3.1 Overview and Architecture

Our approach for estimating the number of passengers in a vehicle can be considered a specialized variant of the smart phone tracking approach described in [6]. The basic idea is that WLAN-enabled mobile devices are periodically sending so-called *probe requests* as part of their IEEE802.11 protocol operation to detect the access points that are present in their surroundings. In order to completely cover the frequency spectrum during their scans, the devices typically

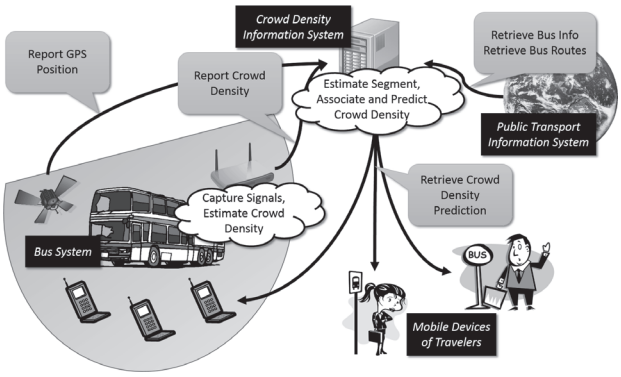

Figure 1: System Architecture

repeat their probe request on all available channels. Thus, given adequate network monitoring hardware, it is possible to overhear these request by simply tuning into one of them. Moreover by continuously monitoring the presence and absence of the probe requests, it is possible to accurately count the mobile devices that are in the vicinity of the network monitoring hardware.

Once the number of passengers has been estimated, it needs to be made accessible to the travelers. To do this, it is first transmitted to a central server where it is then combined with the associated segment of the current route of the vehicle. To compute this association, we rely on the positioning information provided by the vehicle itself by means of a built-in GPS receiver. We then combine with the static route information managed by the public transport operator with the GPS position to determine the current route segment that the bus is traversing. As a last step, we then store the vehicles route segment with the associated crowd-level and a timestamp. Finally, the resulting data is made accessible to travelers which can then retrieve the crowd density estimations for the public transportation system for different times of day through their mobile devices.

The overall system architecture is depicted in Figure 1. It consists of three main components, namely the system inside the vehicle which is responsible for determining the crowd density and capturing the current GPS position, the public transport information system which is responsible for providing geo-spatial information about the routes that the vehicles are operating on as well as a crowd density information system which integrates the information and makes it accessible to travelers. While there are many possible options to split up the responsibility of determining crowd density from WLAN signals, we decided to keep all computations regarding probe requests local to the system inside the vehicle. This means that apart from GPS position, the system solely transfers the current crowd density. The reason for this is twofold. First, this reduces the overall bandwidth requirements when compared to transferring all probe requests to the server. Second, it also protects the privacy of the passengers since the transfered data is hard (and in most cases impossible) to attribute to individual passengers.

In the following, we describe the two main issues, namely the crowd density estimation in the vehicle as well as the vehicle tracking at the server-side in more detail.

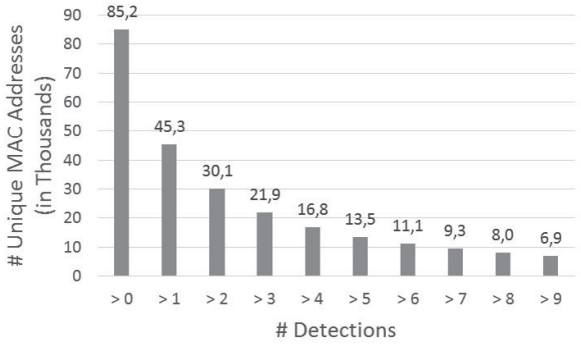

Figure 2: Detected Devices over 14 Day Period

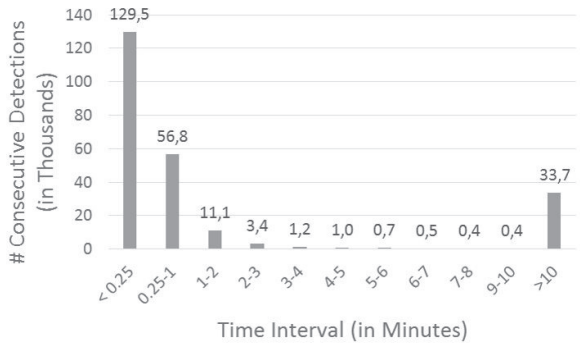

Figure 3: Probe Request Interval Distribution

3.2 Crowd Density Estimation

As indicated previously, our approach to crowd density estimation is based on the idea that WLAN-enabled devices are periodically sending *probe requests* in order to detect the access points that are nearby. In order to completely cover the frequency spectrum during probing, the devices typically repeat their probe request on all available channels. Using a WLAN device that is put into monitoring mode, it is possible to receive the probe requests of nearby devices by simply monitoring a particular channel. By keeping track of the MAC addresses of the devices sending out the probe requests, it is then possible to determine the time duration that a certain device is close to the monitoring device.

When applied to public transportation, an important difference between prior work and our scenario is that in our case, the monitoring WLAN device is a) mobile – since it mounted inside a vehicle – and b) often moving through a densely populated area. As a consequence, we can expect that the monitoring device will not only receive signals from mobile devices that are located in the vehicle but it will also receive signals from devices that are simply nearby the vehicle. This problem is amplified by the fact that in typical public transportation networks, stops at important locations (e.g. in the city center) are targeted by multiple lines. Thus, when a vehicle is stopping in order to allow passengers to enter and exit the vehicle, passengers waiting for another vehicle from another line will be detected as well.

To demonstrate this problem and to develop a solution for it, we have installed a WLAN monitor in one bus operating in the city of Madrid, Spain during a period of 14 days. During the time, the bus was operated for 224 (out of 336) hours and while it was operating, we logged the probe requests received by the monitor. To avoid duplicate detections of the same requests sent out multiple times, we limited the amount of logged probe requests to 1 request per MAC address per second.

In total, the monitor logged 384874 probe requests from 85212 unique MAC addresses. However, as indicated in Figure 2, from these unique MAC addresses approximately 40000 where only seen once and an additional 15000 addresses were only seen twice. These numbers clearly demonstrate the fact that a significant fraction of mobile devices were most likely not traveling in the bus. Instead, it is more likely that they were located at a crowded bus stop or somewhere close to the street where the bus was driving.

To filter out these MAC addresses, while still being able to report changes quickly, we decided to integrate a sliding window mechanism that would remove addresses that were not detected over a longer period of time. In order to configure the windowing period, we further analyzed the logs to determine the typical rate at which we would detect probe requests from devices.

Figure 3 shows the results extracted from the logs. As indicated, the vast majority of probe requests - approximately 185000 - are transmitted within one minute. From these requests, roughly 12500 are transfered within 15 seconds or less, meaning that they are most likely repeated requests that were not filtered out by our 1s rate limitation. The remaining 60000 requests, however, are sent at least 15 seconds later which indicates that they might be new requests. Looking at the overall slope indicated by the histogram in Figure 3, it seems apparent that the vast majority of consecutive probe requests are heard typically within 1 and at most within 3 minutes. Interestingly, the histogram also shows that there is a significant number of consecutive probe requests that are repeated within an time frame above 10 minutes. However, we attribute these to stationary devices that are picked up multiple times during the 14 day period when the bus traverses routes multiple times.

Given these results, we configure our sliding window mechanism for the crowd density estimation to 3 minutes. In order to avoid the counting of devices that are not within the bus, we suppress devices that have not been detected for at least 1 minute and we continue to count them until their signals are no longer contained in the window - meaning that the WLAN monitor has not received a probe request for at least 3 minutes.

3.3 Vehicle Tracking

Once the crowd density has been estimated, it needs to be assigned to a particular route and segment (i.e. the pair of previous stop and next stop of the vehicle). However, in European cities, estimating the route that a vehicle is taking by simply connecting the different stops will result in a very coarse grained estimate of the route. Instead, it is necessary to model the route by means of a more detailed representation such as a polygonal path that defines multiple waypoints between the stops.

To determine the current location of the vehicle using the possibly imprecise GPS, we rely on basic geometric opera-

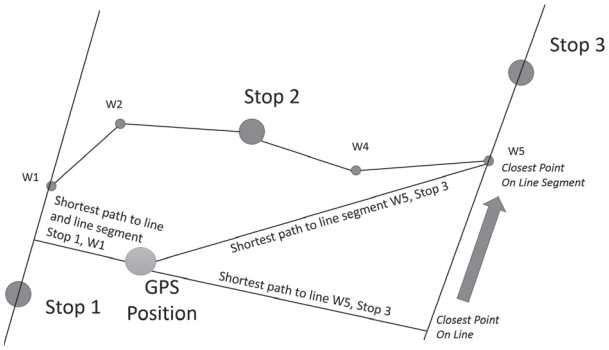

Figure 4: Vehicle Tracking Approach

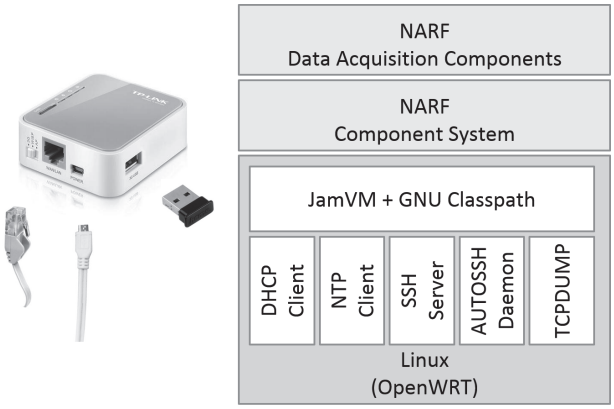

Figure 5: Bus System Hard- and Software

tions on top of an accurately modeled polygonal paths representing the routes. Thereby, the basic idea is to compute the shortest paths to all line segments as depicted in Figure 4. Technically, this is done in three steps. First, we compute the closes point to each line segment of the path. Note that this is either the perpendicular line between the line segment and the GPS position (left) or in cases where the perpendicular line does not intersect within the segment, it is one of the two points defining the line segment (right). Then, we compute the distance between the GPS position and the closest point for all line segments and finally, we use the segment with the shortest distance as the current position on the route which identifies the previous and the next bus stop.

To minimize the computational overhead of the resulting computations in a spherical coordinate system, we simply interpret the GPS coordinates as Cartesian coordinates. While this may result in imprecisions when applied to larger distances, we did not find this problematic at a city level. To test this, we tracked three buses over the course of 2 weeks and verified the validity of the resulting bus stop sequences by comparing them with the route information. In all cases, the bus stop sequences were matching the sequences of the route, however, due to the limited update rate of 2 position updates per minute, some bus stops were sometimes skipped.

4. IMPLEMENTATION

In the following, we briefly describe a number of implementation issues that we had to tackle in order to deploy the system. To put these issues into a meaningful context, we first describe the existing infrastructure before discussing the details of our implementation.

4.1 Infrastructure

The Madrid bus system encompasses roughly 2000 vehicles that operate more than 200 routes. All buses are equipped with WLAN access points that provide free Internet access to the travelers. For this, the access points are equipped with a 3G network card. In addition, all buses are equipped with a GPS system. A central system polls the GPS information from the buses regularly at 30 second intervals. The gathered GPS information is then used to estimate arrival times and to dispatch new buses if delays are detected.

4.2 Bus System

To implement the crowd density estimation inside the buses, we rely on an additional low cost off-the-shelf access point (TP-Link 3020) as WLAN monitor which we equip with a USB memory stick to increase its internal memory for logging purposes. In order to connect the access point to the Internet, we connect it to the existing bus systems (i.e. the existing access point that provides 3G Internet connectivity to passengers). To be able to monitor the WLAN network, we replace the firmware of the device with a custom built of OpenWRT that is tailored to our needs.

Besides from packet capturing support via TCPDUMP, we install a number of system services depicted in Figure 5. To acquire an IP address from the existing access point in the bus, we run a DHCP client. In order to enable remote administration despite the firewall of the 3G network provider, we connect to one of our servers through AutoSSH and establish a tunnel to the device's SSH server. Finally, since this device does not exhibit a real-time clock, we rely on NTP in order to set its clock upon restart.

On top of this, we install JamVM with GNU Classpath in order to execute Java code. This enables us to use the NARF Component System [2] to handle the actual crowd-density measurements. To do this, we rely on existing components from the NARF component toolkit to handle the data transmission and windowing which we extend with a component that taps into TCPDUMP and interprets its output. Since our access point does not exhibit a real-time clock, we configure the device to boot up with its date set to 2012. When the NTP client on the device has successfully determined the current time at least once, this date will be adjusted to the current date (i.e. a date in 2013). In the crowd-density estimation code, we check the current time and suppress all further actions until the time is set to 2013. This effectively avoids stale readings and allows us to buffer crowd density estimations on the device together with a correct time stamp in case that the 3G connection is temporarily unavailable.

4.3 Public Transport Information System

To associate the crowd density information with a particular segment of a bus line, we extend the existing transport information system with 3 web services that expose some of its information. The first web service makes a list of routes available. The second service enables the retrieval of detailed

```
{
  "Id":4281,
  "LineId":17,
  "Loc":
    {
      "Lat":-0.001534102118749,
      "Lon":-7.489303515333618
    },
  "Route":33342
}
```


Figure 6: Bus Information Output Example

route information including bus stops and the polygonal line that connects them. Finally, the third service exposes the real-time information about the current bus location as well as the route that it is operating on.

All web services expose the information as JSON strings which are compact and easy to parse in most programming languages. An example for the bus information output provided by the real-time service is depicted in Figure 6. Besides from the bus id (Id) and current bus location (Loc), the output also contains the id of the bus line (LineId), which reflects the id used by the citizens and a pointer to the current route (Route) which enables the retrieval of the stops and waypoints using the route information web service.

4.4 Crowd Density Information System

The last component of our implementation is the crowd density information system. Implemented as a set of Java Servlets, the system ties together the bus and route information provided by the Public Tansport Information System and the crowd density estimation provided by the Bus System. To do this, it provides a web service that enables the WLAN monitor in the bus to upload its latest crowd density measurements. Furthermore, it continuously polls the Public Transport Information System in order to acquire the latest bus information.

When the Servlets are initialized or when a route change is detected, the system downloads the new route information for the bus and begins (or continues) the vehicle tracking. Whenever a new GPS coordinate for a bus is retrieved, the coordinate is matched against the polygonal path describing the route to determine the current route segment. The route segment is then associated with a timestamp and buffered in memory for future use. When a Bus System performs an upload of some crowd density information through the web service offered by the Crowd Density Information System, the system uses the timestamp that has been assigned on the Bus System when the estimation was created to determine the buffered route segment that corresponds to the reading. The resulting crowd density report for a particular route segment is then stored in a database for later retrieval through travelers.

At the present time, our implementation of the Crowd Density Information System simply provides a map-based visualization of the route information that has been captured over different time intervals. An example for this is shown in Figure 7. The black lines indicate bus routes through the city of Madrid for which crowd density information has been captured. The thickness of the lines indicate the crowd

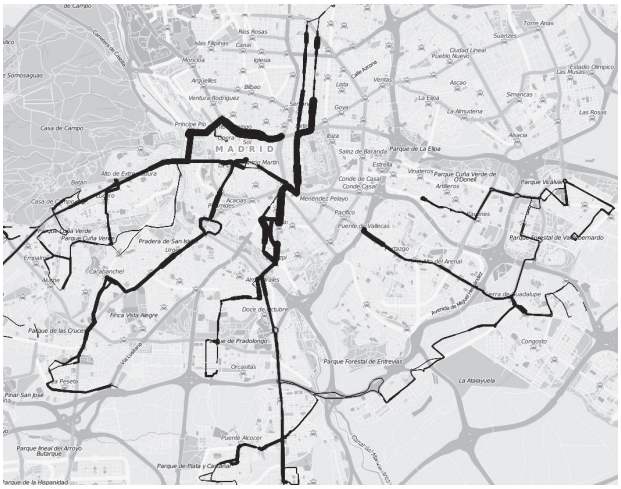

Figure 7: Crowd Density Visualization Example

level for a particular segment of the bus route. As our next step we plan to integrate this information into a mobile bus navigation application for Android devices as part of the prototype development in the GAMBAS European FP7 research project.

5. EVALUATION

In the following, we evaluate our approach to crowd density detection with respect to the design goals identified in Section 2. To do this, we first discuss the system characteristics with respect to *automation*, *cost* and *privacy impact*. Thereafter, we provide an initial report on the *latency* as well as the level of *accuracy* achieved by our system.

5.1 Discussion

As described in Section 2, we attempt on supporting *full automation*, *low cost* while ensuring a *low privacy impact*. Given the approach and its implementation described in Section 3 and Section 4, these design goals are addressed as follows:

- *Full automation:* The presented approach for crowd density estimation is based on overhearing the probe requests that are sent by IEEE802.11 enabled mobile devices. These requests are automatically transmitted by the devices as part of their normal protocol operation. As a result, the approach will work without the installation of any additional software and thus, there is no need for passengers to be actively involved in the collection process at any point. Similarly, due to the integration with the existing services operated by the public transport provider, there is also no need for any manual intervention from drivers or other personnel. Instead, once it is installed, the complete system is fully automated.

- *Low cost:* In order to deploy our crowd density information system, we try to optimally leverage the existing infrastructure - i.e. the 3G connectivity and the GPS receiver - that is already available in the vehicles. However, in order to perform the actual monitoring we

extend the infrastructure with one additional access point. At the time of writing, the cost for the device and the USB memory stick which we are using ranges well below 50 Euros. At the server side, we introduce additional services built on top of J2EE technology. Given the platform agnostic nature of Java, they should be easy to integrate into an existing web-based infrastructure. As a result, we are convinced that the overall deployment cost of the system is reasonably low - especially, when compared to other alternatives such as camera systems, for example.

- *Low privacy impact:* Due to the fact that our system applies passive monitoring of IEEE802.11 enabled devices, it is possible to uniquely identify travelers across all vehicles of the complete public transport system. As a result, the chosen approach can be considered quite invasive from a privacy perspective. To minimize the possible negative impact on the privacy of the travelers, our implementation of the approach is distributed. Instead of collecting all raw messages at a central system, each WLAN monitor is set up to be able to compute a crowd density estimation locally. Once an estimate has been computed by the monitor, it only transmits its id, a global timestamp and the number of passengers in the bus - which is then processed and stored centrally. As a result, we argue that the privacy impact on the user is minimal. Although it may be possible to track individuals in cases where the vehicle utilization is very low (i.e. close to 1 passenger), in cases were the utilization is higher, identifying individual travelers is most likely very hard – if not impossible.

5.2 Experiments

To determine the degree of fulfillment with respect to the design goals of achieving a *low latency* and a *high accuracy*, we have deployed the WLAN monitors in 3 buses that are operating in the city of Madrid, Spain. At the time of writing, these buses have been collecting data for 3 weeks using the approach and implementation described in Section 3 and 4. In the following, we briefly describe our experiences with respect to latency and accuracy.

5.2.1 Latency

Based on the size of our windowing mechanism which uses a 3 minute window in order to determine the density of the crowd, our crowd density estimation approach introduces at least a three minute time difference. However, due to changes in network connectivity of the monitored vehicle, this latency can become temporarily higher in cases where the computed crowd density cannot be transmitted immediately. In order to visualize the probability of such cases, Figure 8 depicts the inter-reporting arrival time differences of the 75985 reports collected by our buses.

Since we configured our monitors to report crowd levels every 30 seconds (which reflects the GPS update interval of the existing transport information system), we would expect that if the vehicles 3G connection is reliable, the resulting arrival time difference would lie around 30 seconds as well. Out of the 75985 reports, 72028 reports (94.7 %) are reported with an arrival time of less then a minute and 75175 (98,9 %) are reported within 1.5 minutes or less. As a consequence, in the vast majority of all cases our crowd density

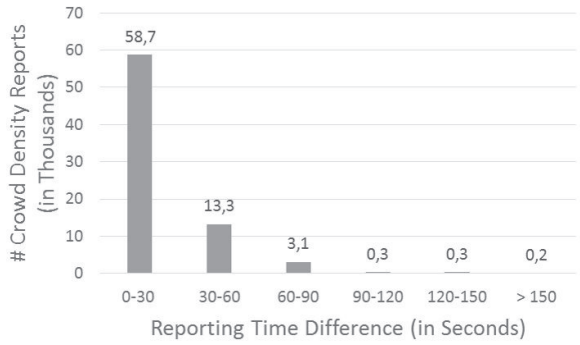

Figure 8: Crowd Density Reporting Latency

reports are available at the Crowd Density Information System within less than 5 minutes. Consequently, we think that the system is broadly applicable from a latency perspective.

5.2.2 Accuracy

In order to determine the accuracy of the system, we performed an initial analysis by means of manual counting the persons in one of our three buses over a 30 minutes trip from the start to the end of the bus' route. After the trip, we compared the reported crowd density measured by our system with the manually gathered information. During the experiment the bus contained between 22 and 52 passengers. Given the total capacity of 65 passengers, the bus was sometimes rather crowded. During the test, the system was able to continuously detect around 20% of the passengers on average.

To put this number in perspective, it is important to note that according to comScore, there are approximately 22.6 million smart phones in Spain[1] and the total Spanish population is estimated around 46.7 million persons[2]. Thus, we would expect that the number of persons captured by our approach would typically level off at around 49%. In addition, several smart phone users may have turned off their phone's WLAN interface in order to save power. Thus, given the rather stable 20% over trip, we believe that the approach can be used to gather reasonable crowd density estimates - however, it is clear that a more extensive study is necessary to confirm these initial results.

6. RELATED WORK

For a traveler two important pieces of information include when the desired vehicle is going to arrive at his/her stop and how crowded it will be. These two pieces of information pose challenges for two separate domains namely crowd density estimation and the estimation of the actual arrival time of the vehicle. For the later, this in turn requires information about the current position of the vehicle over time. In the following we give a brief overview of related work for these two domains.

[1]Number of smart phones in Spain available at: http://www.comscoredatamine.com/2013/01/what-are-the-spanish-doing-on-their-smartphones/

[2]Current estimate of the Spanish population available at: http://en.wikipedia.org/wiki/Spain

6.1 Crowd Density Estimation

Estimating crowd density in indoor and outdoor locations is an active area of research. A number of techniques has been used to estimate the crowd density with high accuracy. These techniques can be mainly classified into image processing and radio frequency based techniques. Some of the work using image processing techniques includes [8],[5], [12],[14] and [4]. [8] estimates crowd density in an outdoor environment by extracting image features using a grey level dependency matrix, minkowski fractal dimension and translation invariant orthonormal chebyshev moments. The extracted features are classified using self-organizing maps. [5] uses pixel counting approach for segmenting the foreground image from the background image and derives and proves that the geometric correction for the ground plane can be directly applied to foreground pixels. [14] provides a survey on crowd analysis techniques based computer vision and image processing. These camera based techniques though reasonably accurate requires careful mounting of cameras in buses such that maximum visual coverage is attained. Moreover, once installed further modifications of their placements is difficult to achieve and thereby is a costly and a time consuming process.

Recently crowd estimation using radio frequency based techniques have gained attention from the research community. Some of the recent work includes [11],[13], [6],[7]. [11] uses the Bluetooth transceivers on mobile phones for estimating the number of people. The approach taken by the authors is based on the assumption that considerable number of people have the Bluetooth transceiver on their mobile phones in discoverable mode. The approach relies on different information such as number of visible devices, links between visible devices, the ratio of number of devices in the current scan to the number of devices in the previous scan, device visibility durations, etc. The authors report to achieve accuracy of more than 75% in their testing scenario. [6] uses a WiFi based solution for detecting and tracking users. The system relies on detecting WiFi probes sent by mobile phones and received by WiFi monitors installed at different places. However, the WiFi probes sent by mobile phones exposes the MAC address of the device which can be used to violate user's privacy. [7] provides an insight on the vulnerability of user privacy because of exposition of such explicit identifiers. [13] uses wireless sensor network based solution for estimating crowd density. The approach employs an iterative process which includes collection and analysis of received RSSI values from the network, construction of training database using K-means algorithm and design of a spatial-temporal stability calibration mechanism to minimise noise. Apart from image processing and radio frequency based solutions there has been some work on using audio samples for estimating crowd density. [3] suggests an audio tone counting solution in which each device (mobile phone) sends a unique tone and at the same time receive tones from other devices. The sent and received tones corresponds to a bit pattern which is then combined to generate new bit pattern. The process continues until the counting is completed.

In our system presented in this paper, we have employed a radio frequency based solution. Specifically our system estimates the crowd level in the bus by keeping track of WiFi probes sent by the mobile phones of users in the bus. In this way our approach resembles with the one mentioned in [6]. However, in contrast to that approach, our system specializes in estimating the crowd density in moving buses which requires filtering of incorrect information when the bus pass through different parts of the city. This incorrect information, in our case are the WiFi probes sent by the mobile phones in the vicinity of the bus.

6.2 Vehicle Tracking

In the recent years vehicle tracking has been the focus of research community. Some of the examples include [15],[1], [9], [11] and [10]. [15] presents a participatory sensing system in which users on the bus share their locations using their mobile phones with a central system which then communicate this information to other users waiting for the bus. The information is then used to predict the bus arrival time. In order to capture the user location the system relies on GSM cell tower information. For the ground truth the bus routes are divided into different segments where each end of segment is marked with three strongest GSM cell towers. The system them matches the GSM cell tower information to which the user is connected to and compare it with the ground truth to predict the location of the bus which in turn is used to predict the bus arrival time. The detection of user's presence on bus is done by detecting the audio beep generated by the ticket checking machines installed at the entrance door of the buses. [1] is a bus tracking and arrival time prediction system. The system requires smart phones to be installed on the buses. Smart phones convey the GPS coordinates of the bus and send them to a back end server. The back end server uses this information and calculates the arrival time of the bus to a particular stop and convey this information to the interested user(s). [9] is also a participatory system which require its users to install an app on their phone. The app serves two purposes, it detects whether the user is in a bus and if yes then it start sending the user's location to a back end server which then computes the arrival time for a particular stop. The detection of users presence on the bus is done by the combination of accelerometer and GPS sensors.

In our system presented in this paper the location of buses is acquired through GPS modules already installed on buses. A GPS module transmits the location of bus every 30 seconds. Our system collects this information through web services offered by the bus transportation company and using the technique described in Section 3.3 calculates the location of the bus between two stops.

7. CONCLUSIONS

Today, most information systems for urban public transportation are empowering travelers to optimize their trips with respect to travel duration. However, solely relying on trip duration as the primary indicator for satisfaction can be limiting. In urban settings providing more information such as the expected number of passengers can be beneficial since it enables travelers to further optimize their comfort. In this paper, we described a scalable and fully automated approach for determining the number of passengers in a vehicle. Furthermore, we discussed our experiences with a deployment of the resulting system in the city of Madrid. Our initial report on the system performance indicates that it can indeed provide a reasonable performance at low cost while preserving the travelers privacy.

At the present time, our implementation of the system

provides a rather simple map-based visualization of the route information that has been captured recently. As our next step, we are integrating the crowd information into a mobile bus navigation application for Android devices as part of the developments in the GAMBAS European FP7 research project. This application will integrate the crowd density estimations directly into the output of a trip planing engine which will enable travelers to take more informed decisions when considering the route and time of a trip. In the long run, we hope that applications like this can help to balance the load on the overall public transport system which – besides from improving the comfort of travelers – could reduce the operational costs of the network.

Acknowledgments

This work is supported by UBICITEC e.V. (European Center for Ubiquitous Technologies and Smart Cities) and GAMBAS (Generic Adaptive Middleware for Behavior-driven Autonomous Services) funded by the European Commission under FP7 with contract FP7-2011-7-287661. The authors would like to thank the remaining members of the GAMBAS consortium for their work on and support for this paper.

8. REFERENCES

[1] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson. Easytracker: Automatic transit tracking, mapping, and arrival time prediction using smartphones. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, pages 68–81, New York, NY, USA, 2011. ACM.

[2] M. U. Iqbal, M. Handte, S. Wagner, W. Apolinarski, and P. J. Marron. Enabling energy-efficient context recognition with configuration folding. In *International Conference on Pervasive Computing and Communications (PerCom)*, March 2012.

[3] P. G. Kannan, S. P. Venkatagiri, M. C. Chan, A. L. Ananda, and L.-S. Peh. Low cost crowd counting using audio tones. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 155–168, New York, NY, USA, 2012. ACM.

[4] V. Kostakos, T. Camacho, and C. Mantero. Wireless detection of end-to-end passenger trips on public transport buses. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1795–1800, 2010.

[5] R. Ma, L. Li, W. Huang, and Q. Tian. On pixel count based crowd density estimation for visual surveillance. In *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, volume 1, pages 170–173 vol.1, 2004.

[6] A. B. M. Musa and J. Eriksson. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 281–294, New York, NY, USA, 2012. ACM.

[7] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall. 802.11 user fingerprinting. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, MobiCom '07, pages 99–110, New York, NY, USA, 2007. ACM.

[8] H. Rahmalan, M. Nixon, and J. Carter. On crowd density estimation for surveillance. In *Crime and Security, 2006. The Institution of Engineering and Technology Conference on*, pages 540–545, 2006.

[9] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson. Cooperative transit tracking using smart-phones. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 85–98, New York, NY, USA, 2010. ACM.

[10] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 85–98, New York, NY, USA, 2009. ACM.

[11] J. Weppner and P. Lukowicz. Bluetooth based collaborative crowd density estimation with mobile phones. In *Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on*, pages 193–200, 2013.

[12] J. H. Yin, S. A. Velastin, and A. C. Davies. Image processing techniques for crowd density estimation using a reference image. In *Invited Session Papers from the Second Asian Conference on Computer Vision: Recent Developments in Computer Vision*, ACCV '95, pages 489–498, London, UK, UK, 1996. Springer-Verlag.

[13] Y. Yuan, C. Qiu, W. Xi, and J. Zhao. Crowd density estimation using wireless sensor networks. In *Mobile Ad-hoc and Sensor Networks (MSN), 2011 Seventh International Conference on*, pages 138–145, 2011.

[14] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.-Q. Xu. Crowd analysis: A survey. *Mach. Vision Appl.*, 19(5-6):345–357, Sept. 2008.

[15] P. Zhou, Y. Zheng, and M. Li. How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 379–392, New York, NY, USA, 2012. ACM.