People density estimation using Wi-Fi infrastructure

Master Thesis

Radoslav Buchakchiev

Aalborg University
Department of Electronic Systems
10. Semester - Networks and Distributed Systems

Electronics and IT
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
People density estimation using Wi-Fi infrastructure

Project Period:
Spring Semester 2016

Project Group:
Group 1023

Participant:
Radoslav Buchakchiev

Supervisor:
Lars Møller Mikkelsen
Tatiana Kozlova Madsen

Copies: 1

Page Numbers: 48

Date of Completion:
June 2, 2016

Abstract:

In recent years, there have been an increase in number of Intelligent Transportation System (ITS) services and applications. In order to realize these services, huge amounts of data about the world is needed. This data should be obtained through infrastructure but deploying it can be costly, and not all cities can afford it. This financial barrier is what holds back the wide adoption of the needed infrastructure. In this work, we design and implement a prototype of a low cost system for people density estimation. This is done by implementing a scanner that collects Wi-Fi probe requests emitted from smart devices. The system is tested in a real life scenario on a city bus. An algorithm is designed and realized that filters the registered probes, specifically with the purpose of estimating the number of people on the bus. The algorithm is further improved by taking RSSI of the probes into account. From the results, it is concluded that the algorithm can estimate the number of people on the bus, with some limitation in accuracy. Taking the low cost of the proposed system into account, it is very promising. Several points for further research are suggested that possibly could improve the accuracy of the estimation.

Contents

Glossary

3G	Third Generation.
AP	Access Point.
EU	European Union.
GPS	Global Positioning System.
IEEE	Institute of Electrical and Electronics Engineers.
IP	Internet Protocol.
MAC	Media Access Control.
OFDM	Orthogonal Frequency-division Multiplexing.
RFID	Radio-frequency Identification.
RSSI	Received Signal Strength Indicator.
RTS	Request to Send.
TCP	Transmission Control Protocol.
USB	Universal Serial Bus.
WSN	Wireless Sensor Networks.

Preface

This master thesis is written on the 10^{th} semester of the program Networks and Distributed Systems at Aalborg University. The work done during the semester was a lot of fun - from doing the research for the problem to doing the tests. This master thesis made me realize that even something looks simple, there is a lot of thoughts and work behind it. I believe the experience learned from the last semester is invaluable for myself. I learned that nothing is as one's expects and there are always problems along the road to success. Working alone requires full responsibility for the outcomes of my actions. When I felt lost, my supervisors were always available for help. The work with them was a pleasure, and it was one of the reasons that motivated me to do my best. I am satisfied with the outcome of the thesis, and I will carry the feeling of accomplishment with me for a very long time.

Aalborg University, June 2, 2016

Radoslav Buchakchiev
<rbucha13@student.aau.dk>

Chapter 1

Introduction

Today we live in a highly technological world. More and more of our surrounding is going "smart". We have a smart TV, a smartphone, smart light, smart watch. Technology makes our life easier, better, high productive. One new trend of building or improving cities is called a smart city. It uses intelligent devices to make the use of city assets like electricity, water or city services e.g. transport and traffic system more efficient. For example, a smart traffic light can provide better traffic throughput and reduce the waiting time. Thus, people will drive less time on the road and cars will produce a lower amount of carbon dioxide. Also on some routes, there are programmable message signs that can change the speed limit depending on the traffic or give information if there is an accident down the road. Another example is a recent development of applications and systems that show traffic on the roads in real-time or use the traffic information to guide the user to the destination thus improving the user experience. All these examples need data to function correctly and to offer high quality of information.

We can see the impact of these systems on some cities. However, the problem of further spread and deployment of this systems in other places is the amount of money that it is needed to spend to build system infrastructure. Most of the countries, cities or municipalities simply do not have the finances to build the infrastructure. It is costly to start from scratch or much effort to integrate already existing systems. Therefore, alternative solutions have to be searched for example using existing Wi-Fi or cellular infrastructure.

One particular service that gains attention these days is people tracking and movement. Person localization is the heart of the new enhanced version of 112 emergency telephone number. Along with the phone call, a location of the origin of the call is sent to the emergency operators so help can come faster. Another example is tracking people in a museum. When they are in different sections of a museum or front of an artifact, location aware guiding system provides relevant information for the place or the item.

It is possible to track people with various degree of success using different technologies e.g computer vision[20], RFID[29], wireless sensors[47], infrared[41], ultrasound[16], Bluetooth[21],[42],[32] or GPS[28].

[20] survey several techniques for people counting. The methods exploit machine learning algorithms. They are feasible to count people, but most of the time the algorithms are computationally intensive, and new hardware(video cameras) has to be deployed. For some, it is necessary to calibrate the algorithm on site.

[47] use wireless sensors networks to estimate how many people are in a room based on received RSSI of a sensor signal. The approach is an iterative process in two steps - detection step and calibration step. In the detection step, crowd density is divided into different levels depending on the RSSI data obtained by WSNs using K-means algorithm. In the calibration step, noise and other deviations estimations are eliminated based on spatial-temporal correlation of crowd distribution.

[29] use RFID reader and reference tags. The Euclidean distance is calculated from the received signal strength from the tracking tag and the reference tags. The location of the tracking tag is estimated using K-nearest neighbors with different weights.

[41] use infrared sensors for coarse-grain indoor localization. Every person has to wear a badge that emits unique identifier. The information is received from infrared sensors in the room and then it is sent to a central computer where the location is estimated. The system works well but has problems when there is a sunlight because of the nature of the infrared.

[16] provide high location accuracy using ultrasound time-of-flight technique. In each room on the ceiling, there is a sensor that measures the time of travel of the ultrasonic pulse. After that, the data is sent to a central controller which performs the lateration computation. The system needs a network of sensors which is not very scalable.

Another approach for people counting is using Bluetooth. Bluetooth devices send discovery requests periodically to find the devices nearby, and if a device is not in ”hidden” mode, it will send discovery reply. [21] put scanner device in buses to capture the requests and uses MAC address as a unique identifier. The data is used to create origin/destination matrix to improve bus utilization. [42] use the same approach using Bluetooth, but to count the attenders of a European soccer championship.

Finally, GPS is another method to estimate people density. [23] use crowdsourcing to build its database with GPS logs for density visualization in a city. There was also an android application[28] to help people to find a place to go tonight. While [43] use GPS location traces to help police to assess crowd conditions during city-scale mass events.

All these examples show us that it is possible to estimate people location or density. Cisco visual networking index[4] predicts that 53% of the IP traffic in 2019

will be generated from Wi-Fi devices. Also, the number of Wi-Fi hotspots is growing each year from 64.2 million in 2015 to 435.2 million in 2020. This prediction leads us to consider Wi-Fi infrastructure as a promising method for people density estimation. This report is going to focus on how this is possible.

1.1 Related work

In the previous paragraphs was mentioned that it is possible to track or locate people using different technologies such as computer vision, RFID, wireless sensors or Bluetooth. This section will mention works that use Wi-Fi technology as a matter of estimating the number of people in a given area. As wireless adoption gains huge success and Wi-Fi access point deployment rises scientists are trying to use off-the-shelf network cards and current Wi-Fi infrastructure. Current research falls into three categories regarding localization using Wi-Fi - RSSI, channel state information(CSI) and Wi-Fi probe requests/responses.

RSSI is a common performance metric for [46], [40], [6] and [45].[45] use the RSSI from Wi-Fi AP beacons as a method of sensing how many people are in a room. The idea is because of obstructions and multipath affect the strength of the received signal and depending on how many people are in the room the RSS will be different. Two machine learning algorithms are compared - linear-regression and support-vector regression. The paper finds the correlation between the RSSI and the number of people in a room. For that purpose mobile application is developed and trained for different scenarios. The accuracy of the method is more than 75% chance of right estimation. [6] idea is that people affect the transmitted signal with blocking the line of sight or as a source to scatter the signal. The experiments are done using a pair of Wi-Fi cards with directed or omnidirectional antenna in an indoor and outdoor area. The tests are evaluated for estimation between one and ten people. The outcome is a probability distribution of RSSI as a function of the number of people which is the base for estimation using Kullback-Leibler divergence. The experiments show a good accuracy of the estimation with an error up to two people. [40] use Wi-Fi signals to track people in a queue. Unique features are extracted from a signal trace that describes the status of a human in the queue. The tests simulate people waiting in a queue. A person carries a smartphone connected to a Wi-Fi AP. When she moves toward the AP, the RSSI of the connection gets stronger. The RSSI goes to its maximum when the person is in the service area (e.g. cash desk). When the person leaves the service area, the RSSI suddenly drops. [46] approach is similar to [45], but the test area is instead of one room an apartment with five rooms. Moving average and moving variance are used for detection. To track a single person, a passive radio map of the area is created. Creation of the radio map is a time-consuming process, and it is working only in a static environment. If there is a pet or change in the placement of the

furniture the map will not be valid.

Several papers [33], [44], [39], [22] use Channel State Information(CSI) instead of RSSI. There are Wi-Fi cards on the market that can give OFDM subcarrier information for calculation of channel frequency response. CSI is more sensitive than RSSI to environment changes and can provide insight about signal multipath. [33] employ per-subcarrier frequency response as features of location and rely on machine learning algorithms to classify a device to one of several trained spots. The method achieves over 90% mean accuracy across 50 spots and less than 7% false positive error. The drawback is that the algorithm has to be trained in a new place before it is used.

[44] propose Percentage of nonzero Elements(PEM) metric - a monotonic relationship that relates CSI variation to crowd number. The experiments conclude that the novel method is accurate, scalable and reliable. [39] goal is to identify different activities classified as in-place, walking or tracking for a single person. For each activity and place, CSI profile is created. For in-place activities such as cooking and sleeping the distribution of the CSI amplitude is taken into account, whereas for walking - sequences of CSI amplitude. The profiles may change in time, so it is possible for user feedback to calibrate them. The tests are done solely with one person and assume static environment. Since the system is designed for a single person, the presence of another person or pet will invalidate the CSI profiles. [22] present a design and implementation of SpotFi system for indoor localization. It achieves a median accuracy of 40 cm by finding the angle of arrival(AoA) and time of flight of each multipath component and calculates the most likely location of a target by observed RSSI and estimated AoA.

Recently research was focused on the Wi-Fi MAC layer instead of the physical player. The way that a mobile device searches for nearby APs, with Wi-Fi probes, is exploited to track people, estimating trajectory or finding their density. In order the approach to work, a person has to carry a Wi-Fi capable device, which means people without device or Wi-Fi turned off are not counted.

[32] try to estimate crowd density and people flow at 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. Three different approaches for counting a device are suggested - naive(just counting MAC addresses from a probe request), time-based(consider the time of capture of a probe for flow determination), RSSI-based(a probe is taken into account if the RSSI is above a threshold) and hybrid(combination of the previous two). The conclusion is that it is possible to estimate a pedestrian flow, but external information is necessary for reliable tracking using Wi-Fi probes.

[15] use Wi-Fi probes to measure people density in a public bus system. Since the estimation of the density is done on the router, the user privacy is maximized. A complete software and hardware implementation is designed using low cost off-

the-shelf hardware and platform agnostic software. The architecture of the system comprises of three parts - a system in the bus responsible for people density in the bus and the bus location, a public transportation system that supplies with geo-spatial information about the bus routes and a crowd density information system that integrates the information from the previous two systems and present it to the users. A Wi-Fi router with Wi-Fi card in monitor mode is installed in the bus. The router is pre-installed with OpenWRT[30] so software that can calculate people density can be installed. The router is connected to already deployed Wi-Fi AP in the bus. The transportation system is extended with three web services to supply with bus routes, bus stops and route segments so the density information can be associated with a particular road segment. The system achieves its goals - low cost, user privacy and sufficient accuracy.

[27] show that it is possible to track and estimate a trajectory of a moving device. The tracking is constrained only on streets and roads on a map. The tracking estimation is done using hidden markov model. The paper suggests different ways to increase the frequency of the probe requests send from devices such as RTS frame injection or AP emulation so the trajectory estimation can be improved. The results from the experiments conclude the system scores good accuracy with less than 70 meters mean error and possibility of large-scale urban deployments.

Finally, there are two commercial implementations of user tracking and counting - Cisco Meraki CMX(Connected Mobile Experiences) Location Analytics[5] and Blip Systems[34]. Both of them uses Wi-Fi probe requests. 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.

1.2 Use cases

In section 1.1 we saw that Wi-Fi infrastructure is a promising solution to user tracking and counting. Here several use cases are suggested for which crowd tracking could be useful.

1.2.1 Road congestion

Rush hours in big cities are a major problem. The cause is because the roads were not designed to carry such load of cars. Most of people use the same popular streets and the shortest routes possible. Knowing which roads are congested, the driver can choose a different route to avoid the heavy traffic. This information will help the driver to arrive faster at the destination, use less fuel and thus save carbon emissions.

1.2.2 Shop management

Most of the time shop owners do not have a full picture of people visiting the shop. The receipt log can help what the customers buy and which are the most popular products, but it is hard to find when are the peak hours or is there need for more employees to serve the customers. The owner of a shop is interested how many people are visiting the shop and potential for new clients. Wi-Fi probe requests can show how long customers stay in the shop, revisiting customers or people that pass by regularly. This information could be used to create a distribution of the visitors in the shop during the day and help to optimize the staff availability so on the peak hours there will be enough employees. This improvement will eliminate long queues and increase the user experience.

1.2.3 Public transport

Public transport is an energy-efficient alternative to car-driving because a bus or a train can carry much more people simultaneously. Unfortunately, most of public transport is underutilized which makes it almost the same cars regarding energy efficiency[24]. The information of which buses are underutilized and the most used routes can help a public transport company to make better decisions to reduce the buses on less popular routes and add new buses when it is necessary.

1.2.4 Social activities

Friday night is a great time to go to a bar or restaurant. Usually, people want to go to a place where everyone is going out, or if they are new in the city they want to find where are the popular places. People density estimation information can be used to create a map of ”hot” locations in a region in real-time. Historical data could provide trends of popular places and also show where people go for after party.

1.3 Problem statement

GPS does not work in buildings because satellite signal can not penetrate building’s walls. However, it gives very accurate localization in open areas. The drawback is it demands a mobile device to be awake thus it drains the device’s battery faster which is not wanted by the end-user. Wi-Fi, on the other hand, is turned on almost all the time, because of ”always online” application demands such as Skype, Facebook or WatsApp. GPS is supported mostly on middle to high-class mobile devices while Wi-Fi connectivity is assumed as a basic feature. This knowledge leads us to use Wi-Fi as a cheap, available and low-power consumption method to localize people. While not everyone has a mobile device with Wi-Fi card or Wi-Fi

turned on, using Wi-Fi can give us a good approximation of the number of people in an area. So our problem statement is:

Given a number of mobile devices, how many we can detect in a given time frame?

1.4 Scenario

This report was inspired by the city of Wolfsburg need and will to cooperate. Since 2013 Wolfsburg AG and WOBCOM[1] have provided free Wi-Fi access to all citizens and guests of the city. More than 5000 people have taken advantage of free internet connectivity. In the past, Wi-Fi was used as a feature at the airports, cafes or shops, but today it is expected as a human right. In order to improve travel experience, transport companies started to deploy Wi-Fi AP for their vehicles. With the lower prices of GPS receivers, it started to be installed in trucks or buses for knowing their location in real time and tracking. There are also web services that utilize GPS data and give information for congested streets or when bus will arrive. People can use them to plan their trip and save time.

Most of big cities in the world already provide such features for their citizens, but not yet Wolfsburg commune. However, there are plans to install Wi-Fi and GPS in the public buses, and the situation is going to change. We took the opportunity and pleasure Wolfsburg commune to cooperate with us to make buses mobile sensors for people density. Right now there is Wi-Fi infrastructure that supplies data for statistics such as how many users use the system, which is the most used(”popular”) AP or the internet traffic that is generated. Collecting data from connected users is useful, but gathering information also from all other is much more fruitful.

When a user wants to connect to a wireless network, she first chooses one from the available ones. Operating systems such as iOS and Android provide Personal Network List(PNL) functionality. PNL are list with Wi-Fi networks that the user used them in the past. Next time when the users device is near from a network save in the PNL, it will automatically associate and connect to it. There are two ways devices with Wi-Fi network card find the wireless networks surrounding them - by listening of AP beacons or by sending Wi-Fi probe requests(see section 2.2).

Wi-Fi probe requests are unsolicited by the user and happen all the time with different frequency. If we intercept them, we can use them as user presence. The number of requests received at particular area will be the area density.

Chapter 2

Analysis

In this chapter, 802.11 wireless network preliminary is presented. Next, the purpose 802.11 probe request frame and its implications for the user privacy are introduced. Finally, a people estimation algorithm using probe request in a bus is defined.

2.1 802.11 wireless network preliminary

The discovery of radio waves made wireless communication possible. It allows two or more devices to communicate from a distance without using wires. Through the years, the wireless communication has been evolving regarding speed and quality. ALOHAnet was the pioneer of long distance communication between the Hawaiian Islands. In 1997, the 802.11 IEEE standard defined a specification for a wireless computer network. Two years later, 802.11b-1999[18] amendment and the establishment of the Wi-Fi Alliance[2] lay the foundation for future market compatibility and mass adoption. Nowadays, Wi-Fi is the preferred way for the most people to connect to the Internet.

Networks based on wireless are fundamentally different from the wired one. Instead of using wires and electronic signals, the communication between two devices is made using antennas and electromagnetic waves. Like the Ethernet, radio communication uses shared medium and everybody compete for access to the medium. The radio link is unreliable and heavily depends on the environment. The radio signal can be reflected, scattered or attenuated from obstructions. It also attenuates when it travels from a long distance.

The statements from the previous paragraph also apply for Wireless Local Area Network(WLAN) [1] defined in the 802.11[17] standard. The standard defines Basic

[1]From now on WLAN, 802.11 wireless network or just wireless network are used as synonyms in the text. WLAN is the most popular type of wireless network. If another wireless network is mentioned in the text, it will be named explicitly. Wi-Fi and 802.11 are also used interchangeably.

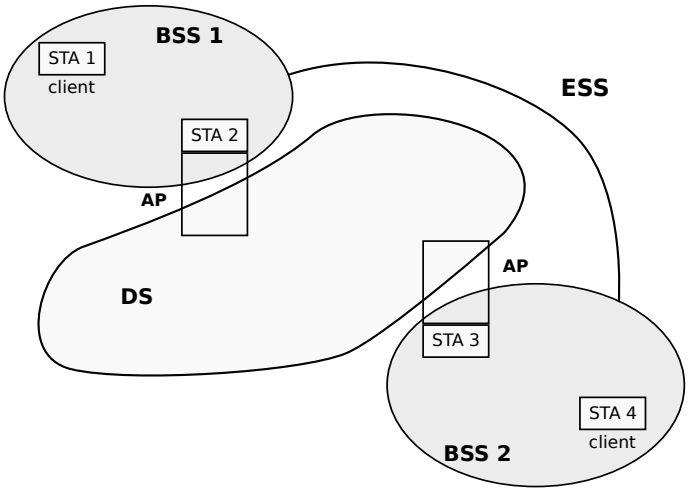

Figure 2.1: Example of Infrastructure BSS network

service set(BSS) as a basic building block of an 802.11 wireless network. BSS is simply a group of stations - devices equipped with a Wi-Fi network card. Each BSS is identified by a 48-bit unique identifier and has a human-readable string named as service set identifier(SSID). There are two types of Wi-Fi networks according to the standard - infrastructure BSS and independent BSS(IBSS). Figures 2.1 and 2.2 give an example of both kinds.

In an infrastructure BSS, there are two types of stations - client and access point(AP). AP functions as a bridge between wireless and wired medium and relays the frames from clients. Clients are mobile stations such as notebooks or cell phones. Two clients do not communicate directly. Instead, they use AP for frame forwarding and delivery. If two or more BSSs share the same SSID and are connected through distributed system(DS), they form an extended service set(ESS) - a one bigger logical wireless network. The DS today is implemented by network switches and distribution medium, which is a backbone network used to relay frames between access points. In almost all commercial solutions, Ethernet is used as the backbone network technology[10].

The other type of wireless network, IBSS, is the smallest possible network that can consist of only two mobile stations. In contrast with infrastructure BSS, it does not have AP or DS. In BSS, nodes communicate directly. IBSS network is often called an ad hoc network. In most scenarios, it is short-lived network created for a specific purpose for example a conference meeting where the attendees want to documents with each other.

The communication between two stations in a BSS is done by transmitting MAC frames. The IEEE 802.11 standard defines three types of frames: data, control and management. Figure 2.3 shows the format of a MAC frame. Data frames

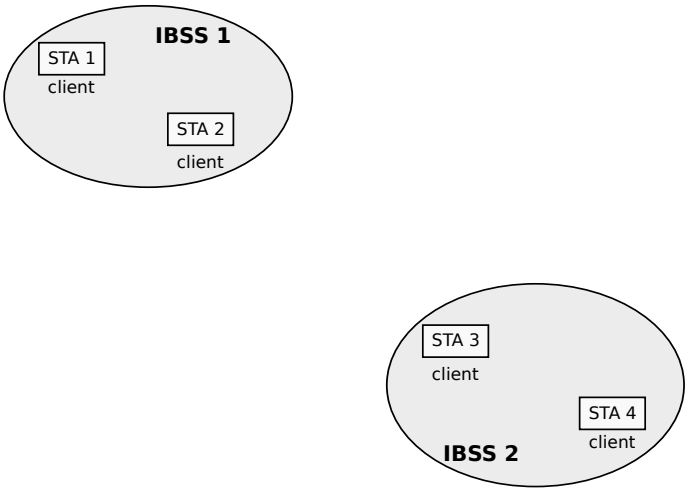

Figure 2.2: Example of two IBSS networks

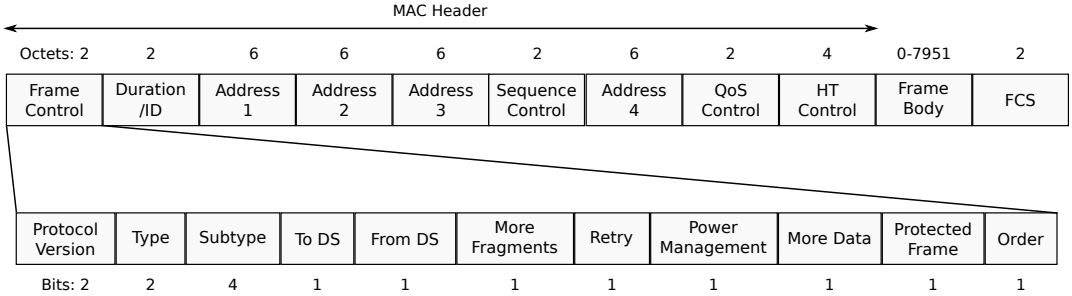

Figure 2.3: 802.11 MAC frame format

carry application data, and their purpose is to transfer data from one place in a network to another, while control frames are used for administration the access of wireless medium. Because of the shared medium, two stations can not send frames simultaneously. When a station wants to send a frame, it first listens the medium to be ”idle”. After each transmission, station waits for an acknowledge(ACK) control frame. It signals a successful reception from the receiver. If ACK frame is not received after a certain amount of time, a collision is assumed, and the frame is retransmitted.

In an infrastructure network, a station needs to be associated with an access point, before it starts to use the services provided by the network. The station sends an association request management frame that carries the supported data rates by the Wi-Fi card and the SSID of the network which it wants to associate with. If the AP accepts the station, it replies with a positive association response. After a successful association, the station needs to be authenticated if the network requires credentials.

2.2 802.11 probe requests

Before a station associates with an access point, it needs first to find it. Beacon and probe request management frames[2] help to solve the access point discovery problem. Beacons are unsolicited by clients and are broadcasted from an AP on regular intervals. Beacon frequencies can vary among APs, but the default one is 100 ms[11]. A beacon carries valuable information about the network that AP manages, such as SSID, supported data rates, authentication type or timestamp for the stations in the BSS.

A client has to listen for a certain amount of time to find the AP near itself. If a node does not want to wait for beacons, it can send a probe request frame. Sending a probe request, a node asks the question "Is there any AP around me?" and waits for a response.

There are two types of probe requests - directed or broadcast. If an AP receives a broadcast probe request, it answers with a probe response management frame the SSID of the network it manages. If a node wants to check if a particular network is around it, it sends a directed probe request with SSID for the interested network. An AP answers with a probe response only if the SSID in the probe request matches with the SSID AP manages. Directed probe requests are the only way to connect to a network when the AP does not broadcast beacons. When a node sends a probe request, it starts a timer for waiting responses. When the timer is over, the node analyzes received probe responses and finds how many and which networks are around it.

Both discovery approaches have advantages and disadvantages. When a device listens for beacons, it saves its battery power, because it does not transmit any frames. However, the discovery process can be delayed from nonfrequent beacons. On the other hand, the probe request approach wastes energy for a transmission of a probe request, but in return, it gives immediate answers from the APs. Sometimes the APs around a device can be many, and the processing of all requests can take a lot of power. If a device wants to conserve battery power, it can send a directed probe request for a saved network from the past. If it receives a probe response, the device can continue with an association or otherwise to send a broadcast probe request for potential new networks.

The standard does not define when or how often probe requests have to be sent by a station. Because of that, the frequency of the probes varies by the network card, driver, mobile device, operating system and application used. Cisco CMX Analytics[25] provides information about Wi-Fi clients and their presence by detecting probe requests sent from them. Table 2.1 presents probe frequency statistic based on mobile phone state. [9] do laboratory tests for several brand of mobile phones with different operating. The results support the Cisco CMX Analytics

[2]Referred in the text also as probe requests or just probes

Device State	Probe Request Interval (smartphones)
Asleep (screen off)	$\sim$ once a minute
Standby (screen on)	10-15 times per minute
Associated	varies, could require the user to manually search for networks

Table 2.1: Frequency of the probe requests based on phone state

statement about probe frequency.

Probe requests are unsolicited, and users are unaware when the phone does send requests. There is no indication from the user interface. As we see from the Table 2.1, the phone sends probes regardless of its state. Recently, researchers found that probes could threaten user identity and location privacy. The next section discusses this problem.

2.3 User privacy impact of using 802.11 probe requests

The original 802.11 standard defined Wired Equivalent Privacy(WEP) as a method of wireless security. It provides data confidentiality as it encrypts the transmitted data frames. Security researchers found major security weaknesses in the WEP design and the RC4 encryption algorithm and 802.11i standard was developed in 2004 by IEEE to fix the issue.

Commercial implementations of 802.11i come in two variants - Wi-Fi Protected Access(WPA) and Wi-Fi Protected Access II(WPA2). WPA uses the same encryption algorithm as WEP, in order to be backward compatible with the old hardware, while eliminates WEP design flaws. On the other hand, WPA2 uses AES encryption algorithm, which is thoroughly tested by security scientists. Even though today WEP is deprecated, and WPA2 is widely used, the encryption is still applied only on data frames. The other two types of frames, control and management, are still transmitted unencrypted which means everybody can capture the association and probe request frames. The last couple of years, researchers found that this protocol design can be used for malicious activity or unveil privacy information about the owner of the device that transmits probe requests.

Using directed probe requests, [3] develop a methodology that uncovers information about the owner of a device like a nationality, often visited places or social relationships with another owner of a device. [7] de-anonymize the geographical region or the city people participating in big events come from.

Not only the directed but also the broadcast probe brings privacy concern. The link between probe, device and owner is a powerful mechanism for people tracking. Interception of a probe request sent from a device in an area can be thought as a presence of the owner of the device in that area. Analyzing the probes can underline people's often visited places, how long they stay there or

daily routes. This knowledge can be used as a base for marketing or surveillance campaigns.

Finally, each device sends probes with different frequency. [8] finds out that the driver for various Wi-Fi cards can be accurately identified by the probes send by the card. The knowledge of the device driver can give a possibility of an attacker to launch driver-specific exploit against a potential target.

Several attempts are made to mitigate the privacy issues of the probes. The base of all of them is to break the link between the MAC address and the device. Two papers, [19] and [14], suggest fixing the problem from its root by introducing disposable identifiers or MAC pseudonyms respectively. Unfortunately, these improvements are not standardized by IEEE, and there is no implementation in any Wi-Fi device on the market. There exists two commercial attempts by Apple in its latest operating system, iOS 8 and also by Google in Android 6[12]; the phone sends probe requests with random MAC address. The Apple's implementations were analyzed by researchers and was critiqued that MAC randomization is based on certain conditions to happen[26]. It is also easy to spot which MAC addresses are anonymized because the implementation generates MAC addresses unassigned by an organization[3]. It is also possible by careful inspection of the sequence numbers of probe requests to link the random MAC with the real one[9].

At present there, is no effective way of successfully preventing the privacy issues of the probe requests. The only sure solution is for the users to turn off the Wi-Fi on their device and turn it on when it is necessary.

2.4 Probe analysis and algorithm

The previous section described the basic concept of a Wi-Fi network and one specific type of frames utilized by such networks, namely probe requests. This section defines an algorithm that estimates the number of passengers in a bus by using probe requests sent by devices.

Clarification: Although it is possible a person to carry more than one device with Wi-Fi turned on, for this report we assume that a person carries at most one device. Since a MAC address uniquely identifies a device, we use device, MAC address and a person carrying a device with Wi-Fi turned on(or just person for short), interchangeably through the report depending on the context. When there is ambiguity between people without Wi-Fi device, it will be mentioned explicitly. Passenger is used to describing a person/device in a bus.

[3]Each organization that develops products using Wi-Fi must register to IEEE MAC Address Block Large

Listening, capturing and analyzing probes can provide valuable information about a presence of a device in a given area and also how long it stays there. By installing a probe capture device in a bus, it is possible to detect people that travel by the bus and eventually to find when they get on and get off the bus.

Before we define an algorithm for people estimation in a bus, we have to think of how the probe trace will look like for a device inside or outside the bus. While the bus is moving, the capture device will collect probes from the devices in range of its Wi-Fi card. Most of the devices will be outside the bus in dense city environment, and their probes will be seen for a short time. In contrast, the devices in the bus will travel along with the capture node and in consequence, there will be more capture probes from them. By collecting probes, the capture device will create probe trace for each device.

From the probe trace of a device, we need to extract the time its owner gets on and the time it gets off the bus. Before we do that, we have to detect if the device is on the bus or not. We define two parameters for people estimation algorithm - *present interval* and *active interval*. The *present interval* is defined as the time needed to pass from the first probe seen from a device in order the device to be decided traveling by bus. Probes from a device are generally sent at random intervals, but during different phone states, the frequency of the probes is almost constant[4]. People in a bus usually use their phone to pass the time so that we can think of the probe transmission as frequent. If a device does not send probe requests not very frequently, it is defined as inactive. We define the maximum allowed time between two probe requests for a device to be considered as active interval. The time between the first and last probe in a trace that satisfies the active interval requirement is defined as the travel time of a device.

Figure 2.4 and Figure 2.5 illustrate how the intervals work and when a device is decided by the algorithm to be inside or outside the bus respectively. The present interval always starts from the first probe from a device, while the active interval is updated with each probe. For a device to be counted in a bus, there should be a probe after the present interval and within an active interval.

Due to the random nature of the probe transmission frequency a device can be marked as in the bus, then outside the bus and again inside the bus on a single short(10-15 minutes) bus trip. We called it a device flapping. The device flapping will not be looked further in this report and will be left for future research.

Another possible inaccuracy is an estimated travel time. The Wi-Fi communication has long range, up to 90 meters. This long range means probes from a far away device can be collected from a capture device on a bus and be estimated to be on the bus. As seen in Figure 2.4, the device is physically outside the bus(the owner left the bus, before the last probe), but the algorithm still thinks it is on the bus.

[4]see Table 2.1

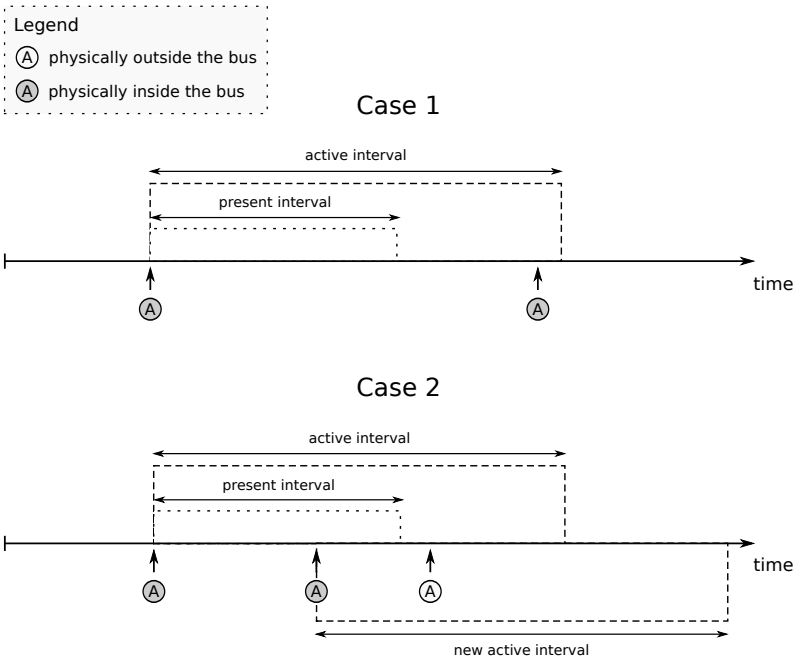

Figure 2.4: Two simple cases of a device(named ”A”) sending probes over time. In both cases the device will be counted as a passenger with estimated travel time between the first and the last probe. In the first case, only two probes are needed. In the second case, the second probe updates the active interval for the device.

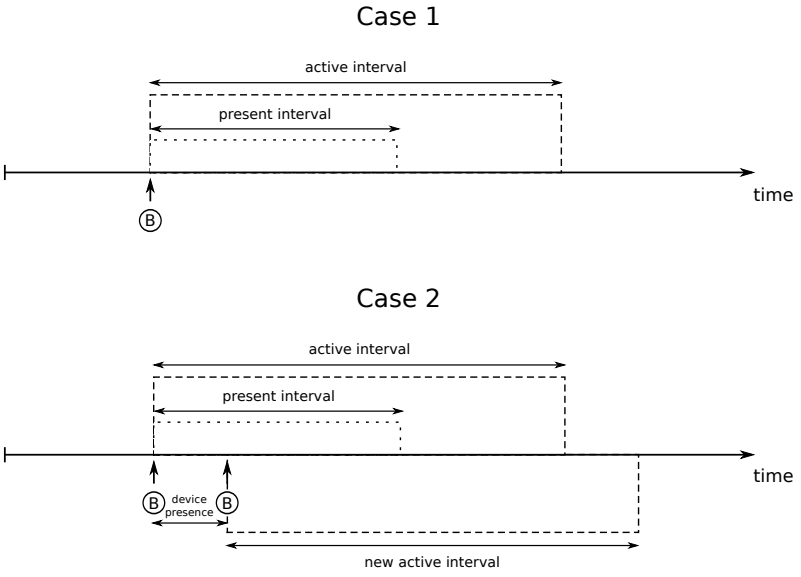

Figure 2.5: Two simple cases of a device(named ”B”) sending probes over time. In both cases the device will not be counted as a passenger with estimated travel time between the first and the last probe. In the first case, there is only one from the device. In the second case, the device is present less than the present interval.

The algorithm works with probe requests, and its performance depends on the penetration of Wi-Fi capable devices, how many people with them use public transport and the bus surroundings. If the bus is moving in highly dense areas, the chances of collecting probes are higher than in rural areas.

Finally, the algorithm uses only the time of reception of the probes and the time between them, but the information carried by the probes, MAC address or network SSID can be used too. The Wi-Fi card driver is also a source of meta information about the probe such as RSSI, channel frequency and data rate. All these pieces of information can be taken into account in future versions of people density algorithm.

Chapter 3

System design and implementation of a prototype

The previous chapter discussed how the Wi-Fi infrastructure could be used to track devices. This chapter will introduce a design and describe example implementation of a system that exploits the probe requests send from a device.

3.1 System vision

This section presents our envision of a system deployed in a public bus transport system that tracks devices using probe requests.

Figure 3.1 shows an example case of probe request scanning by buses. In each bus, a scanner node is installed. The scanner node is Raspberry Pi with a Wi-Fi card in monitor mode. While the bus is driving the scanner listens. The dashed circle around the bus is the reception range of the scanner. On the figure, a mobile phone is depicted as a device that sends probe requests, but it is possible to be a notebook, tablet or other device equipped with a Wi-Fi card. There are different possibilities where a mobile device can be - on a bus, on the road or in a building. In the example, there are phones with turned off Wi-Fi. They are depicted without a "signal arc" above them. It is possible a device to be out of the reception range because it is far away from a bus, signal interference or path obstructions. In these cases, the scanner will not ”hear” them. After a probe request is intercepted it is saved in a database. Periodically the information from the database is sent to a central server where the density is calculated and presented using a web interfaceFigure 3.2.

Counting people based on probe requests has inherent limitation that misses the people without a device. Therefore, the accuracy of density estimation will vary for geographic regions with different levels of mobile phone adoption. Nevertheless, we believe this is a promising approach to exploit Wi-Fi infrastructure

and high grow of the mobile device market.

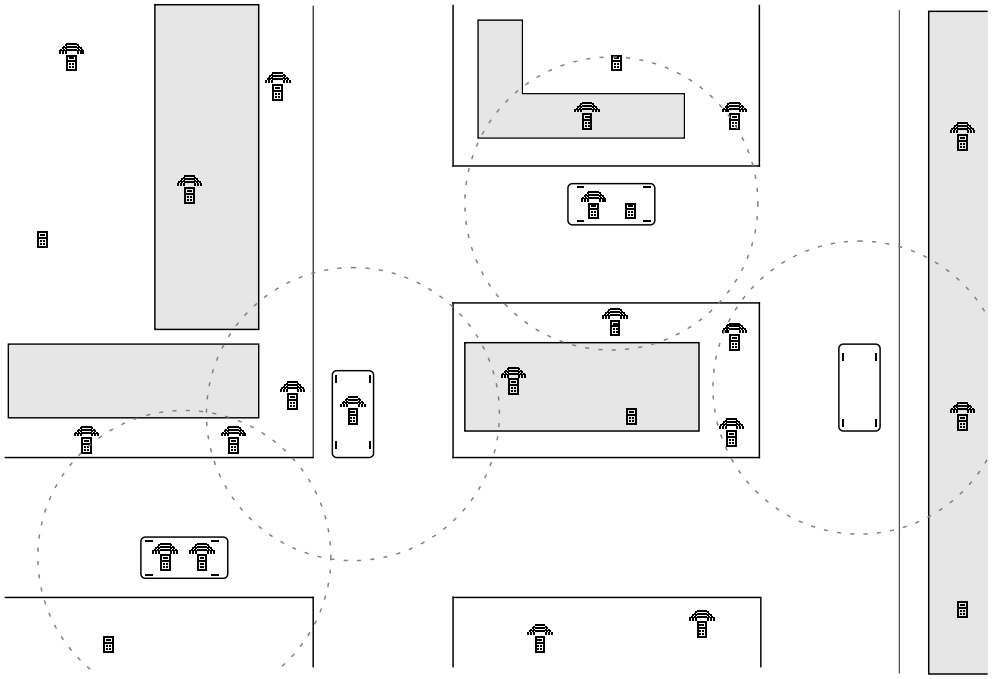

Figure 3.1: An example of intercepting probe requests using scanner device installed on a bus. Dashed circles show the interception range. Mobile phones without an arc above either do not emit probe requests at the moment, or have their Wi-Fi card turned off.

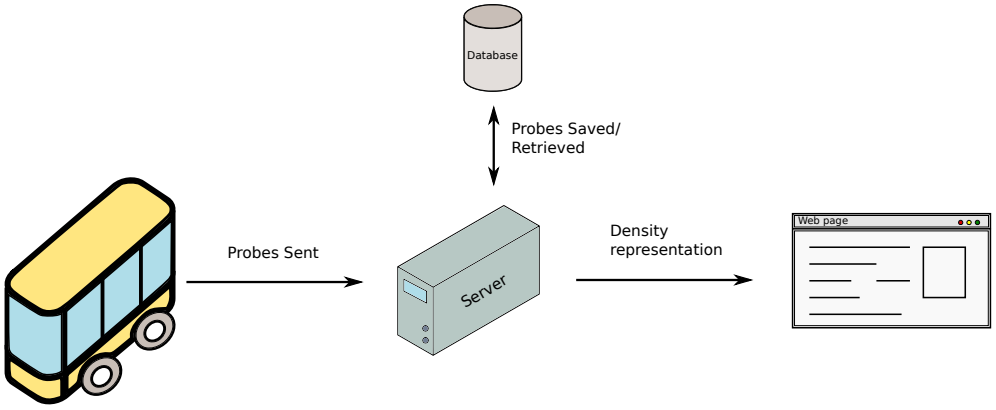

Figure 3.2: Wi-Fi probe travel path: from interception to representation

3.2 System design

Given that, we have a high amount of buses that are constantly moving the system has to be flexible, scalable and maintainable. It has to be flexible so it can be build using various off-the-shelf components, scalable so adding or removing bus will not affect the system and maintainable so it can be easy to deploy and locate and fix problems.

Based on this features the architecture of the system comprises of two type of nodes - scanner and collector. Scanner node is placed on a bus, and collector node is a server machine. Figure 3.3 shows the high-level communication between the nodes. The communication between them is bidirectional with radio waves used as a medium such as 3G. There is one collector node in the system and one or more scanner nodes. There is no message transfer between the scanner nodes; they are independent of each other.

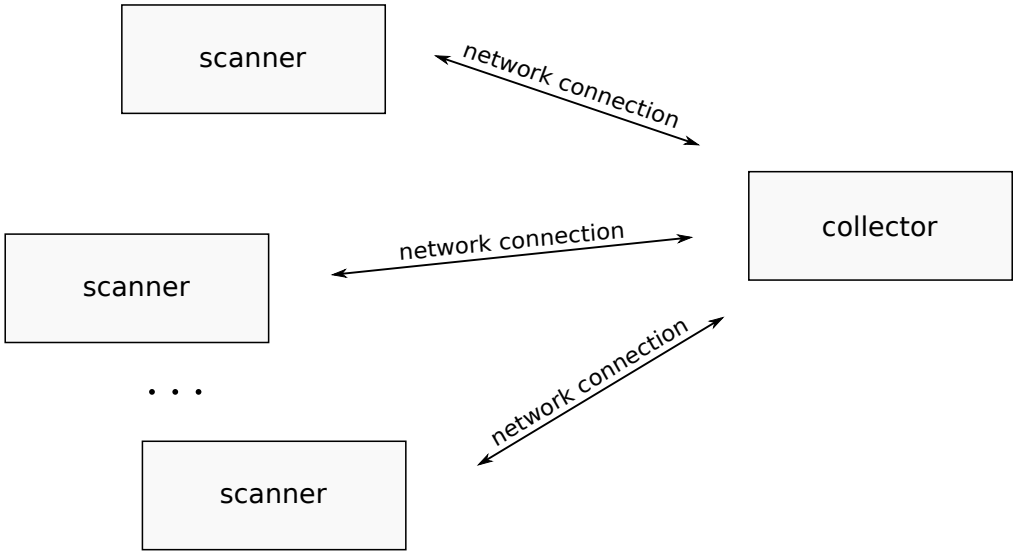

Figure 3.3: Network infrastructure of the system

The scanner node as the name implies scans and records probes send from a device. The sole purpose of it is to gather probes for further analysis. The probes are then forwarded to the collector node to be processed for bus passenger estimation. The functionality of each node is split into several logical modules described next.

3.2.1 Scanner

The idea of the scanner node is that it can be implemented at a low cost, low resource using system. Five modules constitute scanner node - communication,

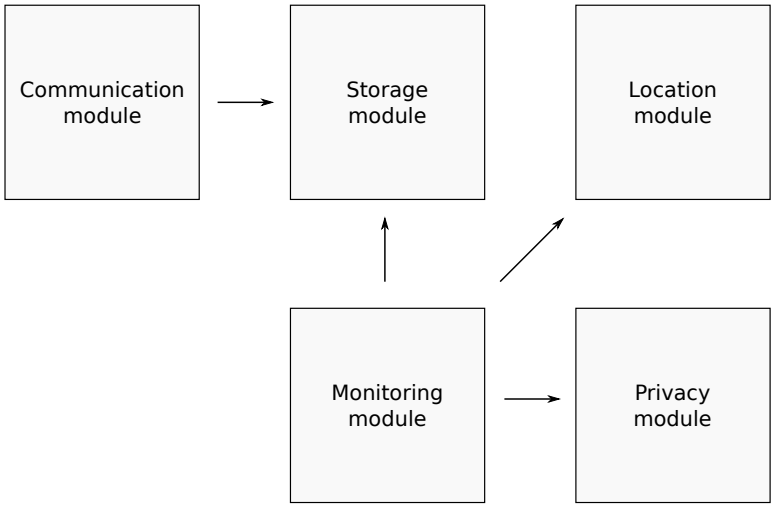

Figure 3.4: Scanner architecture

monitoring, privacy, location and storage. Figure 3.4 shows them and how they interact with each other. An arrow between two modules defines dependency where the module at the start of it uses functionality from a module that the arrow points to. Here is a description of the scanner modules:

- **communication module** - provides the means to talk with the collector and also, authentication and encryption of the messages sent by the scanner. Based on a condition, such as time period or a certain number of captured probes, the probes saved in a database are forwarded to the collector. It is a push based mechanism. The collector is a sink for the messages. This feature provides scalability of the system.
- **monitoring module** - scans and records probes received; the probes are saved using the storage module.
- **privacy module** - scrambles the user identity information in the probe. The EU law[31] does not allow unique identifier such as MAC address to be stored in a database for more than 24 hours.
- **location module** - provides longitude and latitude of the bus. Each collected probe needs coordinates, so the density of a given area can be estimated.
- **storage module** - provides storage(database) so the probes can be saved. It also serves as a buffer if there is network connectivity outage.

To minimize the chances of user data leaks, MAC address anonymization is done on the scanner node. The probes are saved with anonymized MAC address. This way only anonymized data is transferred from the scanner to the collector,

and there is no identity information leak outside of the scanner node. It should be taken into account that in order to track an anonymized MAC address from several scanners, it is necessary the scanners have a way of synchronizing the anonymization algorithm.

A typical data flow starts with monitoring module. When new probe is received, the MAC address is scrambled using privacy module and current location is taken from location module. After that, the probe is saved in the database from storage module. Finally, at some point communication module pick the probes from the database and send them to the collector.

3.2.2 Collector

Collector node is the heart of the system. It collects the probes from all scanners and saves them in a database. Also, it has an interface that can be used to filter the probes and estimate the density of a given area. In contrast to scanner node, collector node role is expected to be served by a high-performance machine that can sustain a big number of network connections has a storage capacity to accommodate the received probes and processing power to calculate area density and respond the queries from the users. Collector node consists of five modules shown in Figure 3.5 that function as follows:

- **communication module** - provides a means of communication with scanner nodes; assure the communication is secure and apply access control so only messages from authenticated scanner nodes can be accepted.
- **storage module** - provides storage(database) so the probes from scanner nodes can be saved in one place and retrieved from the estimation algorithm or serves as an archive for historical and statistical probe data.
- **data processing module** - implements the algorithm described in section 2.4.
- **presentation module** - front end of the controller node. End users interact with the system through this module like sending a request for area density or seeing probe statistics.

There are two data flows. The first flow starts when a message with probes is received and saved in the database using the storage module. The second flow is when a user sends a request from the presentation module. The request is processed by data processing module that asks for probes storage module. After the probes are processed, the result is returned to presentation module and presented to the user.

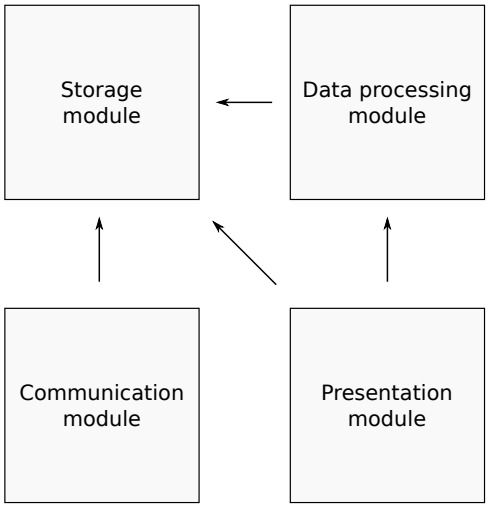

Figure 3.5: Collector architecture

3.3 Implementation

Due to lack of time, the implementation below is just a prototype. Nevertheless, the essential functions of the system are implemented so it can be seen and tested that the approach is plausible. Scanner node is fully implemented except the communication module. There is no network connection between the scanner and the collector node. Only the functionality of the data processing module and part of the front-end user interface are implemented. Full implementation of the system is left for future improvements.

3.3.1 Hardware

Since a scanner node is supposed to be on every bus, the cost of it should be minimal to minimize the total cost of the system. It should have low maintainability, small form factor and deploying it should have no impact on the current bus infrastructure. The bus driver should not and does not have to think about the node in the bus. It should just work.

For that purpose, we use off-the-shelf Raspberry Pi(RPi) hardware as a platform. The model is B revision 1 with 2 USB ports which are sufficient for us to add the functionality that RPi lacks - Wi-Fi and GPS. For probe capture, we need Wi-Fi card that supports monitor mode. The monitor mode of a card allows a device to receive all the traffic from the wireless medium. TP-LINK high gain USB adapter was the choice for packet capture hardware.

Location coordinates, longitude and latitude, are essential for area density estimation. There are two possible cases for GPS receiver placement - external GPS installed in the bus or one attached to a scanner node. Some buses have already

Figure 3.6: Scanner node: Raspberry Pi with WiFi + GPS

installed GPS modules for bus tracking. The GPS information is also used for estimating when the bus will arrive at a bus stop. Since public bus routes span entire cities, bus location is used to locate traffic congestion during the rush hours. Using external GPS receiver, installed already in a bus, means the scanner node will be cheaper. We use USB attached receiver so we can be independent of the bus. However the software that provides access to the GPS information has remote access to and from a software perspective, the GPS placement does not matter. Figure 3.6 shows how scanner node looks like.

Here is exact list of the hardware used:

- Raspberry Pi 1 model B revision 1 with 8 GB SD card.
- TP-LINK TL-WN722N Wi-Fi USB dongle (2.4 GHz).
- USB dongle with u-blox 7 UBX-G7020 receiver chip.

3.3.2 Software

Different operating systems support Raspberry Pi but the most common one is Raspbian[37]. It is free open source linux distribution based on Debian. The software development is based on Python programming language. Python is general purpose programming language that has a gentle learning curve, and it is easy

for making prototypes. There are different python modules for packet capture like pcappy, scapy, pcappy, but pyshark[13] is used, because it is actively maintained, has a low number of dependencies and easy to use application programming interface(API).

GPS receiver control is handled by GPSD[35] software. It provides uniform access for different brands GPS receivers. GPSD is chosen by us because of its two particular features - remote access and reference clock for NTPD[36]. Remote access means that remote machine can connect using TCP to the GPSD and use the receivers maintained by GPSD. This way the GPS placement is irrelevant. The reference clock is very useful because RPi does not have a real-time clock. The device does not keep the time when it is turned off. NTPD(network time protocol daemon) is a software that can synchronize the clock and keep it accurate. It can use as a clock reference another machine over the Internet or GPSD. Since we do not have an internet connection, NTPD is configured to use GPSD as a clock source.

The storage module is implemented using sqlite[38] database. SQLite is serverless, zero-configuration database engine and it has small memory footprint. These features make sqlite ideal for our scanner node. The database schema is defined in Appendix B. We save all possible information for a probe: MAC address, RSSI, SSID for directed probe requests, time of reception and location. There is one field that is not directly associated with a probe - speed. This information is given by the GPS receiver and is the speed of the bus in meters per second. It is not used by the estimation algorithm, but it is included for completeness and eventual future use. There are two meta information fields - scanner id and scanner comment. Scanner id is used by the collector node to distinguish the probes between the scanners and scanner comment is for additional information if needed.

Before probes are saved, they are first captured by the Wi-Fi card. The card is configured to be in monitor mode and listen only on frequency 2412 MHz(channel 1). [9] does an extensible study about the frequency of the Wi-Fi probe request and finds out that more probe requests are captured when channel hopping is not used. The reasoning lies in the fact that the card does not listen the medium when changing the channel. Channels 1, 6 and 11 are nonoverlapping channels in 2.4 GHz band and are most frequent used one. When the phone search for a nearby network it usually sends probes on all channels. We do not know about a survey for most used channels, and we believe choosing channel 1 will not have a huge impact on the tests.

Section 2.3 discussed the user privacy, which is a priority for the design of our implementation. EU also takes seriously about the privacy, and it passed a law[31] that states it is not allowed for a unique identifier such as MAC address to be stored in a database for more than 24 hours. To comply with the european law, MAC address hashing is used. Hashing employs one-way hashing function to transform arbitrary sized argument into a fixed sized output. It is not possible to

reverse the procedure.

Figure 3.7 illustrates the algorithm of hashing a MAC address. The MAC address is combined with a secret value. The result is hashed, and the resulted hash is truncated to 4 bytes. The whole procedure transforms 48 bit MAC address into 32 bit. The output space is smaller than the input, which means there will be collisions - different addresses will produce the same output. The collisions will make two different devices to lock the same, an effect which will lower the accuracy of the algorithm - a sacrifice worth user privacy. A SHA224[1] hash function is chosen in the procedure as it has little complexity and not known collisions. In our implementation of the system, we have only one scanner. We do anonymize the MAC addresses, but no synchronization is implemented in the scanner node.

Until now scanner node implementation was described. The collector node is partly implemented - density estimation algorithm and an early prototype of the front-end. Figure 3.8 shows how it looks like. A map can be divided into tiles with a configurable width(X) and height(Y). The prototype uses probe coordinates from a file; there is no database back-end. The numbers on the figure are artificial and does not represent actual data from tests.

The modifications that are done for the Raspbian distribution can be found in Appendix A.

[1]SHA224 is a member of SHA-2 family cryptographic hash functions

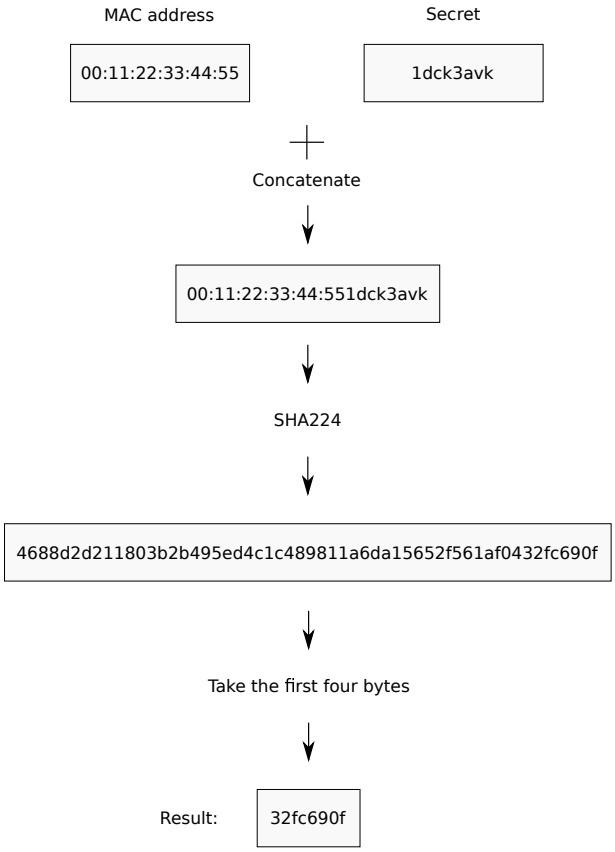

Figure 3.7: MAC address hash algorithm

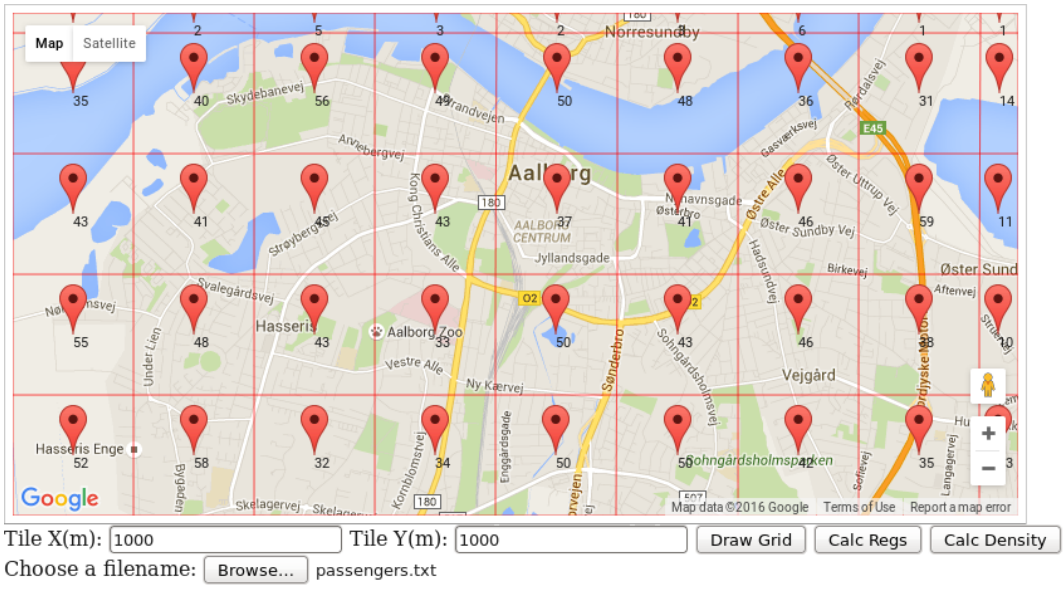

Figure 3.8: Collector user interface(early prototype)

Chapter 4

Tests and runaesults

4.1 Tests

To test the algorithm described in section 2.4, we did two measurements separated by 13 days on a bus from the public transport in Aalborg. Statistics from the two measurements are shown in Table 4.1 While the scanner was collecting probes, we also kept track of the number of passengers manually, in order to compare them to the algorithm's prediction. The manually counted people in the bus is referred as *ground truth*. The resource constraints limited us to do more measurements. The bus route is shown on Figure 4.1 with length 9.59 km. This route was chosen, because of the different surroundings around it. It departs from Aalborg University, then passes through low population density area. After that, it goes near Aalborg train station and passes the center of the city. Because of the variation of the urban environment, we expect to see a variable amount of captured probes in different parts of the route. The bus goes through the whole route about 35 minutes one way. To check the accuracy of the estimation algorithm[1], we also traveled on the bus and counted people going in and out the bus. The time of day of the measurements was chosen to be around 15:30 to facilitate the counting of the passengers, but not to sacrifice the actual traffic of people. Finally, we took measurements from both route ways - from the university to the center and back.

[1]Defined in section 2.4

Date	Probes	Unique devices	Traveler peak
27.04.2016	12720	1662	43
10.05.2016	13471	1667	28

Table 4.1: Measurement statistics

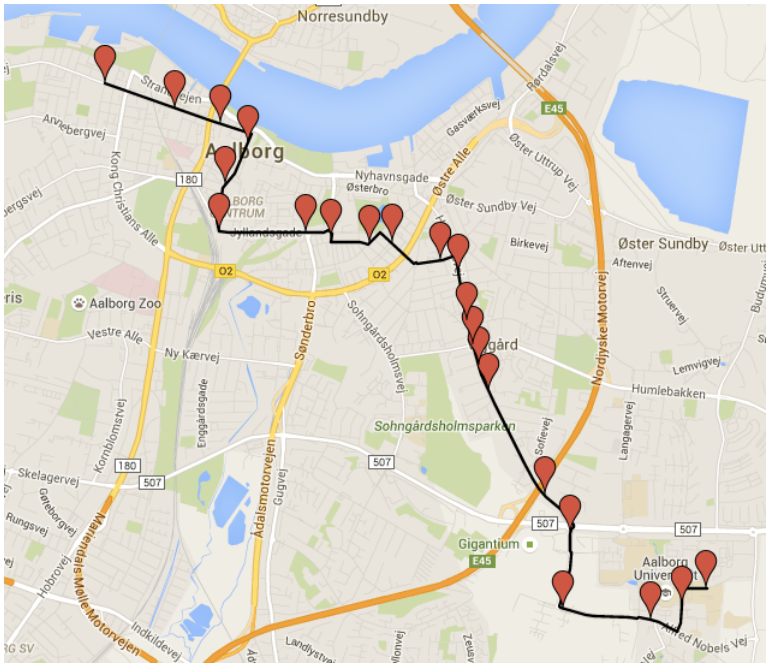

Figure 4.1: Bus route. The red markers are the locations of the bus stops

4.2 Results

This section will introduce the results from the analysis of the collected data. The comparison plots presented in this section are only for the second measurement. The comments and conclusions in the following apply for both measurements.

The algorithm in section 2.4 defines two parameters active and present interval, but it does not say anything about how to choose them. As a guidance, we decided to make a histogram of the difference between two probes from a device. The result is shown in Figure 4.2. From the histogram, it can be seen that the most probe requests are separated by 4 minutes interval. There are some probes separated more than 10 minutes, but we believe this is a cause from a static object near the bus route.

For the active interval, we decided to plot a device lifetime. We define a lifetime of a device as the time between the first and the last probe received by the scanner from a device. Figure 4.3 depicts lifetime of several devices. The shown lifetime is only for the first three minutes from the bus route in order to avoid the clutter from many devices. The overall impression is that the device lifetime varies greatly, and it depends on the device, how fast the bus is moving and the environment. The varying lifetime among different devices leads us that it is hard to put a straight line to say how long has to be the optimal active interval.

The *Unfiltered* graph in Figure 4.4, Figure 4.5, Figure 4.7 and Figure 4.8 is based

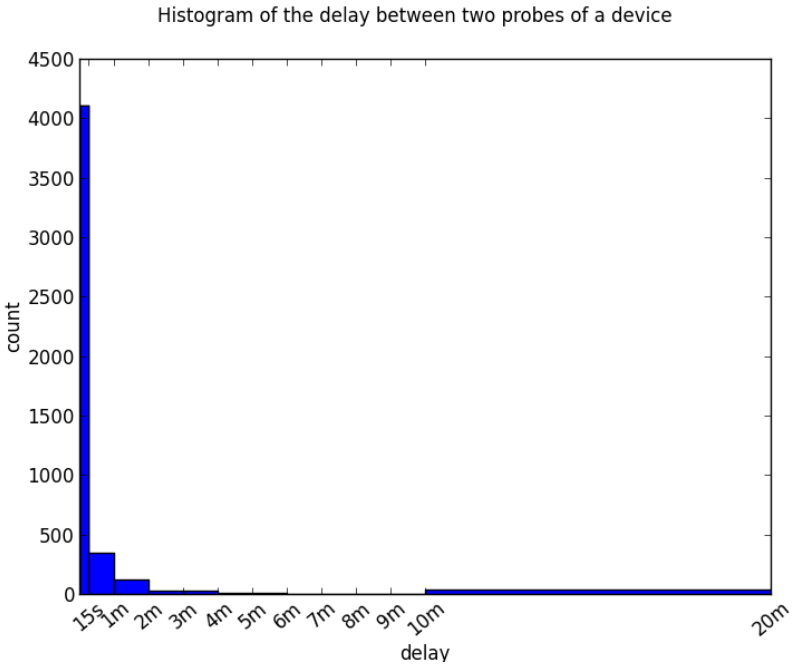

Figure 4.2: Histogram of the delay between two probes from a device

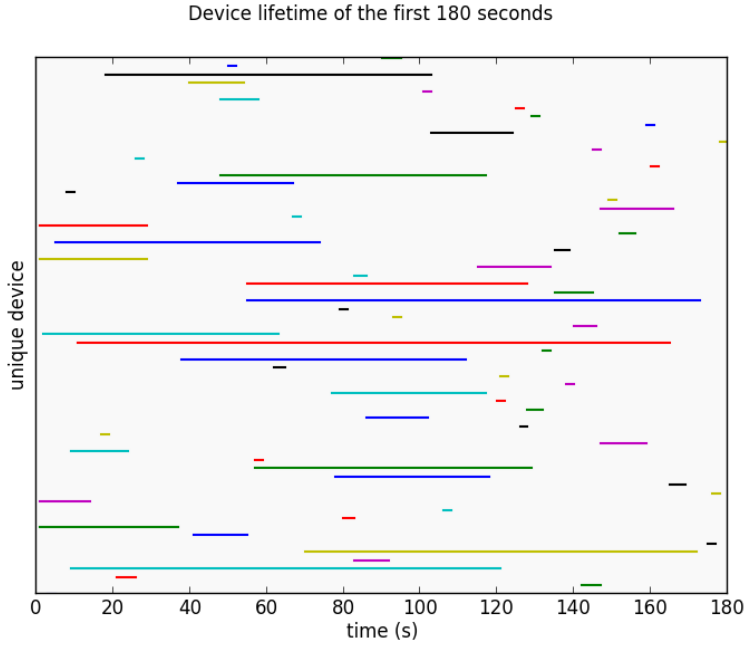

Figure 4.3: Device lifetime of a part of the bus route. Each line represents one device defined as the interval between the first and last probe seen from the scanner.

on the lifetime of the devices. One is added during the lifetime of the device. The value of the Unfiltered graph represents how many devices are ”alive” at a particular second. The graph serves as an upper bound of the estimation algorithm and by comparison with it and by the ground truth it can be seen how good the algorithm performs.

The other parameter, which is the present interval, is a subjective measure. If the present interval is long, it helps to filter the most of the noise[2], but we also miss to count the devices with bus trips less than the interval. For example, if we set the present interval to be 5 minutes, in urban areas where the bus stops are mostly distributed in a 1-minute interval, we risk missing many devices that travel between one and four stops. A long present interval is favorable on long bus rides or when the bus stops are distributed by long distance. If we have a statistic about passenger travel time, it is possible to know more about the consequences of choosing a particular value.

4.2.1 Estimation algorithm

Plots in Figure 4.4 and Figure 4.5 depicts the behavior of the algorithm with varying both parameters. The time axis is the relative time since the beginning of the measurement. The present interval is fixed as 60 seconds in Figure 4.4 because the bus stops on the tested route are distributed around 1 minute, and the active interval is 180 seconds from the insight of the delay histogram. We tested with a longer active interval, but when it is bigger than the maximum time between two probes, the parameter does not have effect anymore.

The shape of the graph matches our expectations - the peak of the graph is where the train station and center of the city is. There are a lot of cafes, and the center is populated with people all through the day.

Varying active or present interval does not do very a good estimation. Half of the time the algorithm estimates more people than actually are on the bus. This estimation means the algorithm is very sensitive to noise and has false positive errors. By comparison with the unfiltered graph, it can be seen that it filters some of the devices, but the overall shape of the estimation curve is the same.

It is important for the estimation algorithm to have lower false positive errors than false negative. We are interested how many people are on the bus, and it is more important for us to know with high confidence that there are a small amount of people in the bus than a high number of people with little confidence. Since not everybody in the bus has a device with Wi-Fi, the estimated passengers are expected to be not very close to the ground truth. The plots proved that using the active, and the present interval are not sufficient for accurate passenger estimation.

We also tested the algorithm with an active interval less than the present one,

[2]Noise is defined as devices that are outside the bus but satisfy the algorithm requirements.

which means that we want the device to send several probes before it is counted to be on the bus. Unfortunately, the probes are sent mostly at random. There is a research paper[9] that does extensive tests on mobile phones for probe frequency and finds that there is a correlation between the mobile phones with different operating system. However, it is not possible in practice to determine a device's brand and model solely based on its probe requests, and therefore this, research cannot be applied directly to our use case.

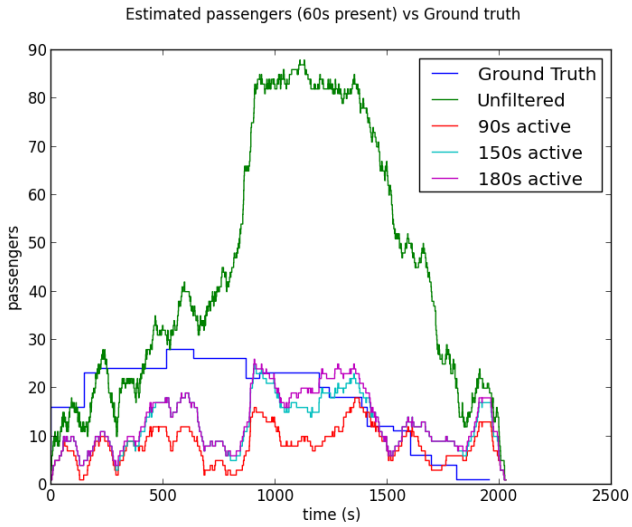

Figure 4.4: Effect on the passenger estimation from varying the active interval

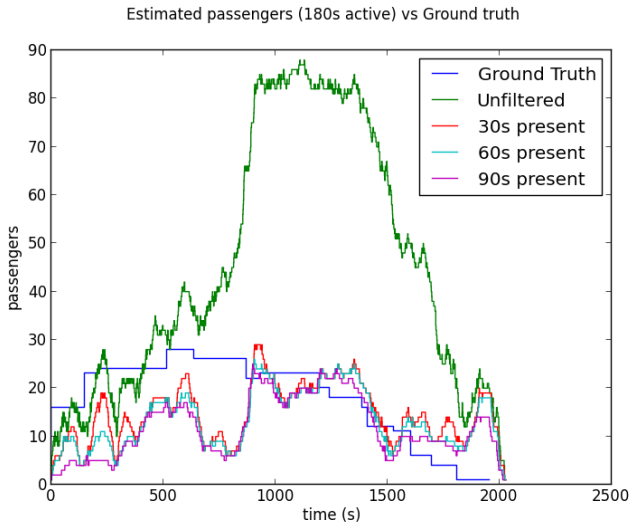

Figure 4.5: Effect on the passenger estimation from varying the present interval

4.2.2 Estimation algorithm with RSSI

The high false positive error of the algorithm defined in section 2.4 driven us to find an improved version of the algorithm. When we did the measurements, we also recorded the RSSI of a probe. The RSSI is often used as an indicator of the distance from the transmitter of the signal. It might be valuable to include RSSI as a third parameter of the estimation algorithm, because if the devices are in the bus or close to the bus, the received probes will have higher RSSI than probes received from devices far away from the bus or in some cases devices with poor transmitter or antenna properties.

Two improved versions of the algorithm(named as base algorithm from now on) are defined that also take into account RSSI of a probe. The first one is using the average RSSI from the probes received from a device. If the device fulfills the active and present interval, the average RSSI from the probes is calculated before it is counted as a passenger. If this average is above a defined threshold, the device is counted as a passenger otherwise is counted as outsider. The second version takes into account the probes from a device that are received with RSSI above a threshold. If the device fulfills the active and present interval, it is counted as a passenger.

To find reference RSSI for threshold, we plot the RSSI of all probes in Figure 4.6. It can be seen that the histograms look like two peaks mountain. The probes with RSSI around and above -60 dBm are likely to be from the devices closer to the bus. On the other hand, the probes with RSSI around -85 dBm are more likely to be outsiders.

Figure 4.7 and Figure 4.8 shows the two algorithms with varying the average RSSI threshold and minimum RSSI threshold. The graphs show conservative number of passengers in comparison to the graphs from the base algorithm. The shapes are different from the unfiltered graph, and there are less estimated passengers during the middle and the end of the trace.

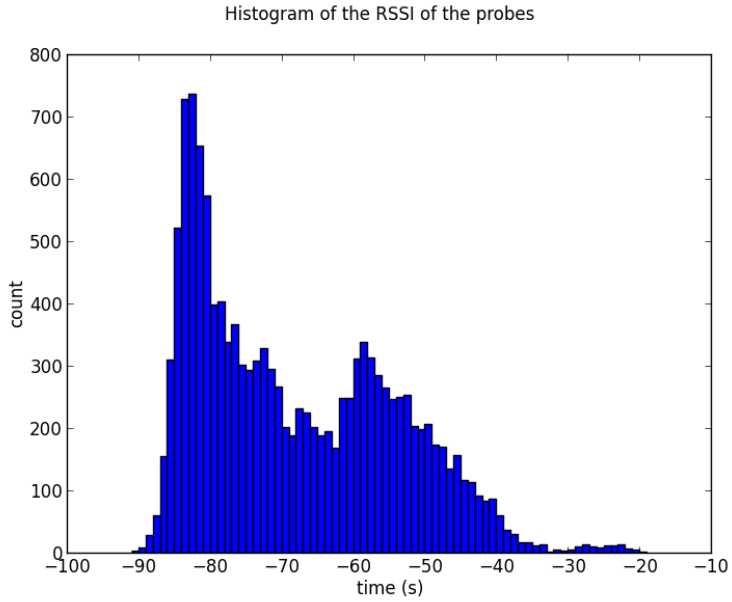

Figure 4.6: Histogram of the RSSI of the captured probes

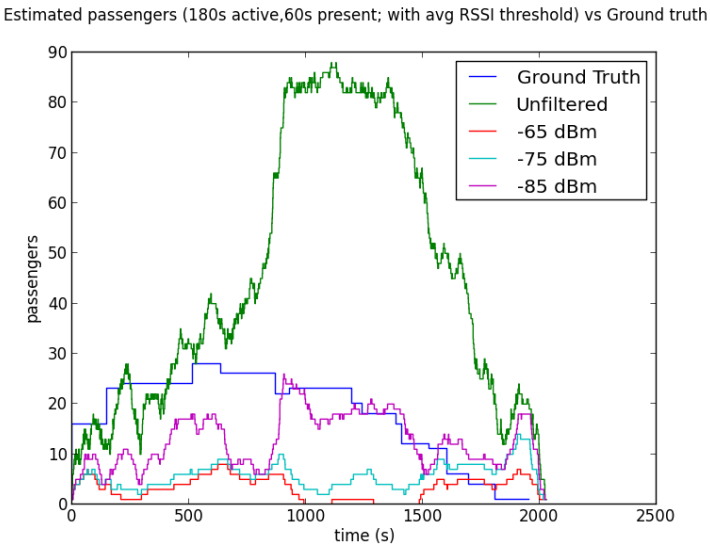

Figure 4.7: Effect on the passenger estimation from varying the average RSSI threshold

4.2.3 Evaluation

We decided to check manually the results from the improved algorithm. We compared the devices that are estimated by it with the actual probe trace. The location of device start and end trip matched near a bus stop which proved to us that the

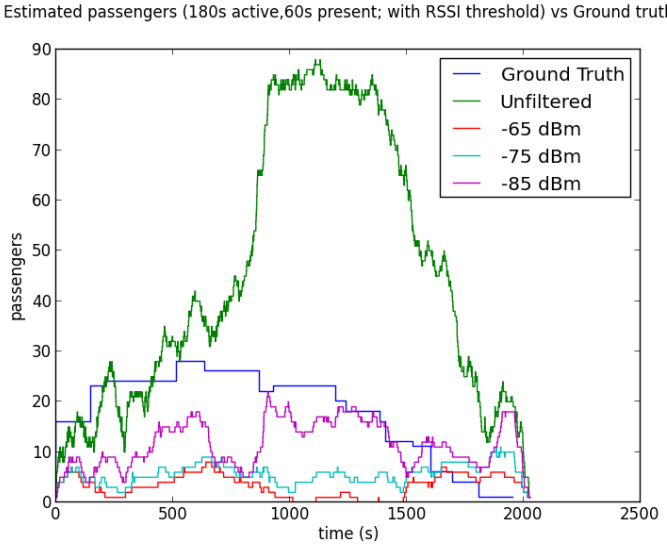

Figure 4.8: Effect on the passenger estimation from varying the minimum RSSI for the captured probes

estimation sounds correct. Unfortunately, We did not have the necessary resources to conduct tests with devices which were known to be on the bus to find how many devices are incorrectly estimated by the algorithm.

In all three versions of the estimation algorithm, there are cases of passenger that we called it as flapping. Some devices that were counted as once on the bus, then leave the bus and later be on the bus again. This situation is possible, due to the random nature of the probes. We noticed that there were less flapping devices in the improved algorithms. We get similar results also when the active interval is really long.

4.2.4 Final notes

In this chapter, we presented how the algorithm defined in section 2.4, which we refer to as the basic algorithm, performs in a real-world scenario. We saw that the two parameters were not sufficient for people estimation, because of the high false positive estimation errors.

Unsatisfied with the performance of the basic algorithm, we developed an improved version that takes into account the RSSI from a probe. We found that its accuracy was better than the basic algorithm and manual checks of the results also confirmed that. The addition of the RSSI in the base algorithm gave us an insight that more information can be included in the estimation algorithm for increasing its accuracy. We are satisfied with the end results and conclude to some degree counting of people on a bus is possible.

Chapter 5

Conclusion

Nowadays technology plays a huge part of our lives. Technology improves the quality of our lives, and makes it easier. Recently, with the fusion of city and technology, a new type of city emerged - a smart city. The smart cities provide a basis for creating new services that enhance the user experience in them. Examples are current traffic situation, finding a free parking lot or the number of passengers and thus the free capacity in a bus. Most of the time each new service requires a new infrastructure to be built for it or integration with a current one for data collection. Building infrastructure and integrating it is a costly process. Most of the cities do not have the finances to invest in these new infrastructures, and thus, alternative solutions should be investigated.

This project took the alternative approach of using current Wi-Fi infrastructure as a source of data and building a service on top it. In chapter 2, the basics of the Wi-Fi network, the Wi-Fi probe request frame and its implications for the user privacy are presented. In the same chapter, an algorithm that estimates people in a bus is defined. Next, chapter 3 portrays a vision of a system that uses probe requests, defines its design and describes its prototype. Finally, chapter 4 discusses the accuracy of the algorithm and two other versions of it.

After building a prototype system using Wi-Fi and testing it, we can conclude that it is possible to estimate how many people travel on a bus with satisfactory accuracy.

5.1 Future work

Counting people using their phones does have its limitations. It counts devices instead of people, which means it will never be 100% accurate. The random nature of the probe requests adds another level of challenge. Here we list several issues, we considered during the report work, but due to time constraints left for future research:

- It is not known when people exactly go in or out a bus. In result, there is a chance the estimated travel time is not accurate. More tests are needed in this direction.
- We can not distinguish which devices estimated by the algorithm are actually on the bus. The manual check that we did is a good way for validation, but cumbersome.
- Phones send probes at not truly random intervals, but periods, based on some device condition. There is no comprehensive statistic about the frequency of the probes among different phone brands or operating systems. This valuable information can give insight about the active parameter of the algorithm.
- It is known how many people have mobile phones in a country, but there is not statistics of the percentage of people that travel with a Wi-Fi device.
- The estimation algorithm is sensitive to different area types such as urban and province. More research is needed to find a correction factor for each area, so the estimation of the algorithm can be adjusted to the actual people traveling on the bus.
- The MAC address randomization has a negative influence on the estimation algorithm. With foreseen mass adoption of the phones supporting it in the future, the accuracy of the algorithm is expected to degrade.

The algorithm defined in this report has two parameters and uses only the time of reception of the probes. It is simple, but effective. The addition of the RSSI in the algorithm proved to be beneficial. The inclusion of the GPS coordinates of a probe and bus stops and use of advanced processing techniques such as machine learning can increase the accuracy of the estimation of people on a bus.

We can find easily how many people are outside the bus, by having a system that is deployed in buses and count people on a bus. By plotting this information on a map, we can create a density map of a given area. Improvements of the estimation algorithm and investigating its behavior can serve as a basis for a people tracking system using a Wi-Fi infrastructure.

Appendix A

Raspbian modifications

Raspbian is a port of Debian linux distribution for Raspberry Pi. From the stock installation, these are the changes that are made to make it as a probe scanner.

A.1 Installed software

Below is listed the installed additional software:

- `sqlite3 - 3.7.13-1+deb7u2`
- `libsqlite3 - 3.7.13-1+deb7u2`
- `ntp - 4.2.6.p5+dfsg-2+deb7`
- `gps - 3.6-4+deb7u1`
- `tshark - 1.8.2-5wheezy18`
- `python-lxml - 2.3.2-1+deb7u1`
- `python-py - 1.4.8-1`
- `pyshark - v0.3.6-22-g9b870d0`

`pyshark` is not a standard raspbian package, and it needs to be compiled from source. The source code is available on `https://kiminewt.github.io/pyshark/`. `python-lxml` and `python-py` are dependencies for `pyshark` that need to be installed.

A.2 Modified files

Here are the files that are changed with their explanation:

- Add udev rules so gpsd can read the /dev/ttyACM0 device and create /dev/gps0 symlink.

  Filename: /etc/udev/rules.d/99-ublox7.rules

```
# Create symlink and change the file permissions
SUBSYSTEM!="tty", GOTO="ublox7_rules_end"

# u-blox AG, u-blox 7 [linux module: cdc_acm]
ATTRS{idVendor}=="1546", ATTRS{idProduct}=="01a7", SYMLINK="gps\%n",\
      RUN+="/lib/udev/gpsd.hotplug", MODE="0666"

LABEL="ublox7_rules_end"
```

  Note that the line that is indented with RUN should be on the same line as the previous one. The line is separated here, because of space constraint.

- Edit /etc/default/gpsd file so the gpsd will be started on system start and find the USB GPS dongle.

  Filename: /etc/default/gpsd

```
# Default settings for gpsd.
# Please do not edit this file directly - use ‘dpkg-reconfigure gpsd’ to
# change the options.
START_DAEMON="true"
GPSD_OPTIONS="-n"
DEVICES=""
USBAUTO="true"
GPSD_SOCKET="/var/run/gpsd.sock"
```

- Edit /lib/udev/gpsd.hodplug so the gpsd daemon started from it can be stopped from the init.d script.

  Filename: /lib/udev/gpsd.hodplug

```
### Excerpt ###
if [ -r /etc/default/gpsd ]; then
  . /etc/default/gpsd
elif [ -r /etc/sysconfig/gpsd ]; then
  . /etc/sysconfig/gpsd
```

```
  GPSD_OPTIONS=$OPTIONS
  GPSD_SOCKET=$CONTROL_SOCKET
fi

# Add pid file option so the init script can stop the daemon
# started from here
GPSD_OPTIONS="$GPSD_OPTIONS -P /var/run/gpsd.pid"
```

- Edit `/etc/network/interfaces` file so the operating system will not touch the `wlan0` interface.

  Filename: /etc/network/interfaces

```
auto lo

iface lo inet loopback
#iface eth0 inet dhcp
iface eth0 inet static
    address 10.11.12.2
    netmask 255.255.255.0

#allow-hotplug wlan0
#auto wlan0
#iface wlan0 inet manual
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf

iface default inet dhcp
```

- Change `ifplugd` config file so it won't touch `wlan0` interface.

  Filename: /etc/default/ifplugd

```
INTERFACES="lo eth0"
HOTPLUG_INTERFACES="none"
ARGS="-q -f -u0 -d10 -w -I"
SUSPEND_ACTION="stop"
```

- Add capabilities `cap_net_raw` and `cap_net_admin` to `/usr/bin/dumpcap` so `tshark` can capture packets as normal user.

  Execute in terminal

```
sudo setcap cap_net_raw,cap_net_admin=ep /usr/bin/dumpcap
```

- Add GPS as a clock reference for `ntpd`. Raspberry Pi doesn't have RTC(real time clock)

  Filename: /etc/ntp.conf

  ```
  # Use GPS as clock reference
  server 127.127.20.0
  fudge 127.127.20.0 time1 0.070 refid GPS stratum 1
  ```

  Note in version 4.2.6 ntpd shared memory driver does not sync the clock if the difference is more than 4 hours. This can be disabled in 4.2.8 version with `flag1` fudge factor.

Appendix B

Scanner sqlite database schema

Here is the definition of the table schema used to store the probes.

```
CREATE TABLE wifi_probe_requests(
    id integer primary key autoincrement not null,
    scanner_id integer, /* Unique id of the scanner node */
    scanner_comment string, /* Placeholder for text */
    mac string, /* MAC address of the sender. Possible anonymized */
    rssi string, /* RSSI of the probe request on scanned node */
    frequency integer, /* Frequency on which the probe was captures */
    ssid string, /* Name of the SSID in the probe request. NULL if broadcast */
    longitude integer, /* longitude location */
    latitude integer, /* latitude location */
    speed integer, /* the speed of the bus when the prob was captured */
    time_of_reception integer /* When the probe was captured */
```

Listing B.1: SQLite table schema for storing probes

Bibliography

[1] Wolfsburg AG. *Wireless Wolfsburg: 2014 weiterhin kostenloses WLAN in Wolfsburg*. http://www.wolfsburg-ag.com/wolfsburg-ag/presse/pressemitteilungen/pressemitteilungen-details/date/30/01/2014/item/wireless-wolfsburg-2014-weiterhin-kostenloses-wlan-in-wolfsburg.htm. Accessed: 2016-04-14.

[2] Wi-Fi Alliance. *Wi-Fi Alliance official website*. https://www.wi-fi.org/. Accessed: 2016-04-25.

[3] Marco V Barbera et al. "Signals from the crowd: uncovering social relationships through smartphone probes". In: *Proceedings of the 2013 conference on Internet measurement conference*. ACM. 2013, pp. 265–276.

[4] Cisco. *Cisco Visual Networking Index*. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html. Accessed: 2016-02-26.

[5] Cisco. *CMX Location Analytics*. https://meraki.cisco.com/technologies/location-analytics. Accessed: 2016-03-03.

[6] S. Depatla, A. Muralidharan, and Y. Mostofi. "Occupancy Estimation Using Only WiFi Power Measurements". In: *IEEE Journal on Selected Areas in Communications* 33.7 (2015), pp. 1381–1393. ISSN: 0733-8716. DOI: 10.1109/JSAC.2015.2430272.

[7] Adriano Di Luzio, Alessandro Mei, and Julinda Stefa. "Mind Your Probes: De-Anonymization of Large Crowds Through Smartphone WiFi Probe Requests". In: ().

[8] Jason Franklin et al. "Passive Data Link Layer 802.11 Wireless Device Driver Fingerprinting." In: *Usenix Security*. 2006.

[9] Julien Freudiger. "Short: How Talkative is your Mobile Device? An Experimental Study of Wi-Fi Probe Requests". In: ().

[10] Matthew S. Gast. *802.11 Wireless Networks: The Definitive Guide*. 2nd edition. O'Reilly, 2005.

[11] Jim Geier. *802.11 Beacons Revealed*. http://www.wi-fiplanet.com/tutorials/print.php/1492071. Accessed: 2016-05-30.

[12] Google. *Android 6.0 Changes*. https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html. Accessed: 2016-05-30.

[13] Dor Green. *PyShark official website*. https://kiminewt.github.io/pyshark/. Accessed: 2016-05-21.

[14] Marco Gruteser and Dirk Grunwald. "Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis". In: *Mobile Networks and Applications* 10.3 (2005), pp. 315–325.

[15] Marcus Handte et al. "Crowd Density Estimation for Public Transport Vehicles." In: *EDBT/ICDT Workshops*. 2014, pp. 315–322.

[16] Andy Harter et al. "The anatomy of a context-aware application". In: *Wireless Networks* 8.2/3 (2002), pp. 187–197.

[17] IEEE. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. http://standards.ieee.org/getieee802/download/802.11-2012.pdf. 2012.

[18] IEEE. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band*. http://standards.ieee.org/getieee802/download/802.11b-1999.pdf. 1999.

[19] Tao Jiang, Helen J Wang, and Yih-Chun Hu. "Preserving location privacy in wireless LANs". In: *Proceedings of the 5th international conference on Mobile systems, applications and services*. ACM. 2007, pp. 246–257.

[20] J. C. Silveira Jacques Junior, S. R. Musse, and C. R. Jung. "Crowd Analysis Using Computer Vision Techniques". In: *IEEE Signal Processing Magazine* 27.5 (2010), pp. 66–77. ISSN: 1053-5888. DOI: 10.1109/MSP.2010.937394.

[21] Vassilis Kostakos. "Using Bluetooth to capture passenger trips on public transport buses". In: *arXiv preprint arXiv:0806.0874* (2008).

[22] Manikanta Kotaru et al. "Spotfi: Decimeter level localization using wifi". In: *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. ACM. 2015, pp. 269–282.

[23] Markus Loecher and Tony Jebara. "CitySense: Multiscale space time clustering of gps points and trajectories". In: *Proceedings of the Joint Statistical Meeting*. 2009.

[24] David MacKay. *Sustainable Energy-without the hot air*. UIT Cambridge, 2008.

[25] Cisco Meraki. *White Paper CMX Analytics*. https://meraki.cisco.com/lib/pdf/meraki_whitepaper_cmx.pdf. Accessed: 2016-02-26.

[26] Bhupinder Misra. *iOS8 MAC randomization - Analyzed!* ihttp://blog.mojonetworks.com/ios8-mac-randomization-analyzed/. Accessed: 2016-05-30.

[27] ABM Musa and Jakob Eriksson. "Tracking unmodified smartphones using wi-fi monitors". In: *Proceedings of the 10th ACM conference on embedded network sensor systems*. ACM. 2012, pp. 281–294.

[28] Sense Networks. *City Sense - Real-time nightlife discovery and social navigation*. https://web.archive.org/web/20160127044313/https://www.sensenetworks.com/products/macrosense-technology-platform/citysense/. Accessed: 2016-03-02.

[29] Lionel M Ni et al. "LANDMARC: indoor location sensing using active RFID". In: *Wireless networks* 10.6 (2004), pp. 701–710.

[30] OpenWRT. *OpenWRT Official Website*. http://openwrt.org. Accessed: 2016-04-10.

[31] ARTICLE 29 Data Protection Working Party. *Opinion 13/2011 on Geolocation services on smart mobile devices*. http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2011/wp185_en.pdf. Accessed: 2016-05-20.

[32] Lorenz Schauer, Martin Werner, and Philipp Marcus. "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*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 2014, pp. 171–177.

[33] Souvik Sen et al. "You are facing the Mona Lisa: spot localization using PHY layer information". In: *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM. 2012, pp. 183–196.

[34] Blip Systems. *Blip Track*. http://blipsystems.com/. Accessed: 2016-03-03.

[35] gpsd team. *GPSD official website*. http://www.catb.org/gpsd/. Accessed: 2016-05-21.

[36] ntpd team. *NTPD official website*. https://www.eecis.udel.edu/~mills/ntp/html/index.html. Accessed: 2016-05-21.

[37] Raspbian team. *Raspbian official website*. https://www.raspbian.org/. Accessed: 2016-05-21.

[38] sqlite team. *SQLite official website*. https://www.sqlite.org/. Accessed: 2016-05-22.

[39] Yan Wang et al. "E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures". In: *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM. 2014, pp. 617–628.

[40] Yan Wang et al. “Measuring human queues using WiFi signals”. In: *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM. 2013, pp. 235–238.

[41] Roy Want et al. “The active badge location system”. In: *ACM Transactions on Information Systems (TOIS)* 10.1 (1992), pp. 91–102.

[42] 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*. 2013, pp. 193–200. DOI: 10.1109/PerCom.2013.6526732.

[43] M. Wirz et al. “Inferring Crowd Conditions from Pedestrians’ Location Traces for Real-Time Crowd Monitoring during City-Scale Mass Gatherings”. In: *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2012 IEEE 21st International Workshop on*. 2012, pp. 367–372. DOI: 10.1109/WETICE.2012.26.

[44] Wei Xi et al. “Electronic frog eye: Counting crowd using WiFi”. In: *INFOCOM, 2014 Proceedings IEEE*. 2014, pp. 361–369. DOI: 10.1109/INFOCOM.2014.6847958.

[45] Takuya Yoshida and Yoshiaki Taniguchi. “Estimating the number of people using existing WiFi access point in indoor environment”. In: ().

[46] Moustafa Youssef, Matthew Mah, and Ashok Agrawala. “Challenges: device-free passive localization for wireless environments”. In: *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. ACM. 2007, pp. 222–229.

[47] Y. Yuan et al. “Crowd Density Estimation Using Wireless Sensor Networks”. In: *Mobile Ad-hoc and Sensor Networks (MSN), 2011 Seventh International Conference on*. 2011, pp. 138–145. DOI: 10.1109/MSN.2011.31.