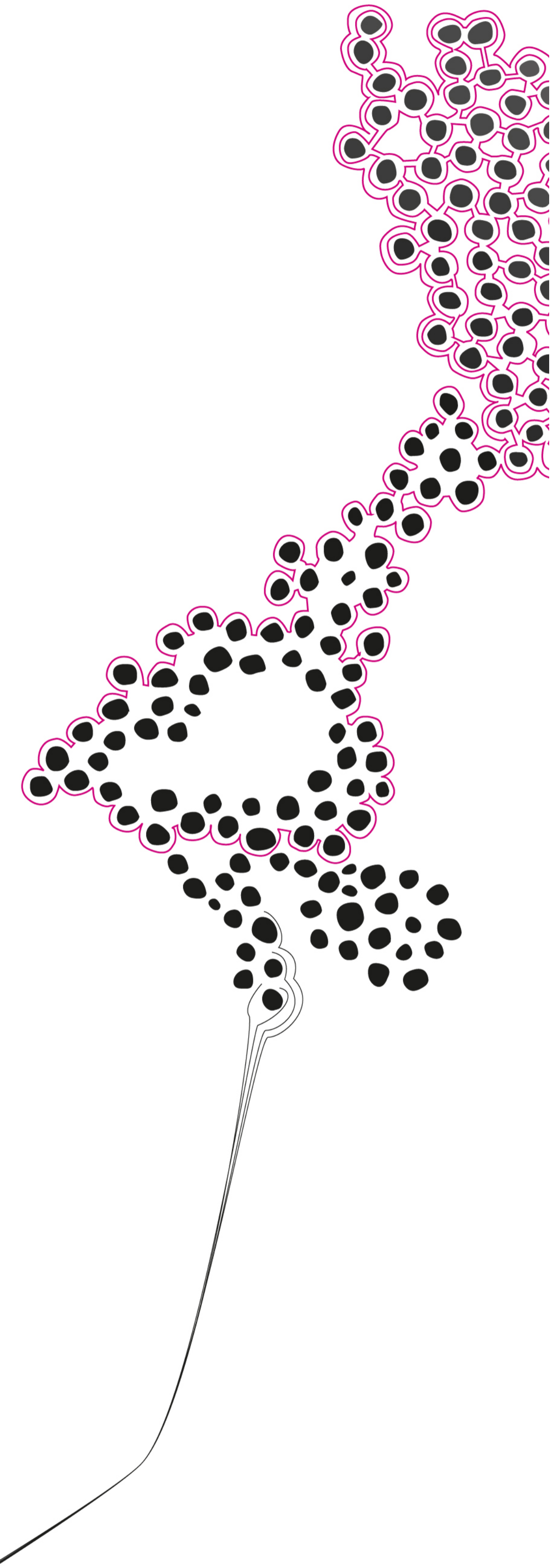


SMART CITIES: CROWD MANAGEMENT USING WIFI BASED INFRASTRUCTURE

By :
K. Braham – s1630083

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
Chair of Pervasive Systems (PS)
Msc. Thesis

Supervised by :
Dr. N. Meratnia
Prof. P.J.M. Havinga

UNIVERSITY OF TWENTE.

MARCH 7, 2018

Contents

List of Figures

List of Tables

Abstract

Smart cities is a modern phenomenon to include ICTs in the development of large urban cities. It helps determining the dynamics of a city by looking at traffic jams and general flow of visitors. Crowd management is one of the key aspects of smart cities, aiding in safety and enjoyable experience for residents and visitors.

The council of Enschede is interested in learning to use WiFi information using the public network maintained by NDIX for crowd monitoring and safety of visitors. This network follows thousands of visitors each day and captures the mobility in the city.

Using this network a model is created to extract the number of visitors in the city, the mobility utilized by visitors and their shopping behaviour as they spend a day in the city. The dataset of NDIX does hide much detail of local motion, as all records are sessions with long time spans. To obtain the ideal situation a reference experiment using a Raspberry PI retrieves WiFi connection status each second. This experiment showcases the possibility of tracking using ideal world scenario.

This work demonstrates the counting of visitors, classification of mobility and suggests improvements for tracking using the WiFi network in Enschede. Results of the events detection based on visitor count is compared with the official monitor of the city council.

1 Introduction

Smart cities is a modern ideology for urban development and integrating Information and Communication Technologies (ICTs) in the city. This framework hugely exploits the power of ICTs to plan, expand, manage and observe the city and its dynamics caused by traffic in the area. One of the pillars of the smart city ideology is crowd management. Crowd management is a topic of interest in smart cities due to the large data pool describing mobility. How do you guide large crowds using modern techniques? Do you nudge them using an incentive to move to a certain area? Do you leave the crowd as is and intervene only for safety related issues? Should you follow everyone and create detailed overviews of all visitors? How do you present information to visitors without overloading them with details of an unknown city?

To answer those questions researchers have developed related work approaches for data gathering, processing and usage in this context. The application varies depending on the final goal of the project. For example, events focus on safety by observing crowd density. This application guides visitors to safe areas, and uses precise local movement to harvest the information. Another application of crowd management is mobility in the city. This approach uses data mining based on city dynamics to classify movement methods, such as walking, cycling or driving, and analyses common patterns. This helps with city congestion and creates options for daily commuters.

In smart cities, crowd management is key to safety and overall experience of residents and visitors. Crowd management ensures proper infrastructure, accessibility, liveableness, social happiness and an enjoyable experience. Smart cities start to play a key role in sustainable urban development. Intelligent systems could alleviate critical problems often seen in growing cities. Important topics as accessibility, sustainability, liveability, noise and economic vitality could benefit from a coherent urban strategy developed using smart systems. These systems have an understanding of the city, learn the dynamics in the city and self-awareness.

The municipality management of Enschede is keen to adapt smart city approaches to their city. It enables new development and insights of the urban environment. Specifically, the municipality management is interested in movement patterns, crowd behaviour and shopping behaviour in the city. The movement patterns depend on the traffic between places in the city, the method chosen for transport and entry points to the city centre. Crowd behaviour focusses on visitor counting, crowd movement, event detection and possibly noise detection in the city centre. Shopping behaviour should give insight in the commercial attractiveness of the city. Do marketing campaigns lead to more visitors? Do visitors walk into multiple shops after the initial shop advertising a product?

This project is a collaboration between the University of Twente, NDIX and the municipality management of Enschede. The goal of this project is to answer questions regarding presence, mobility patterns, shopping behaviour and transport medium in the city centre using the WiFi infrastructure in the city. The WiFi data used is coming from NDIX, which deploys the infrastructure in the city centre. All of the data is anonymised by the providing company before being used in research.

The research questions to be answered in this thesis are:

- *What useful information the WiFi sniffers provide for analysing and observing visitors behaviour and mobility patterns in the city centre?*
- *Is the data sufficiently accurate for counting visitors in specific locations of the city?*

- *How do visitors move within the city centre? Where are the popular places? How people commute within the city centre?*
- *What transportation methods do visitors use when entering the city?*
- *What privacy extensions to the dataset could help minimise personal identification?*

To answer these questions NDIX provides a dataset containing sessions of the WiFi infrastructure in the city centre. A custom build programme will analyse the records and provide information to answer the questions of the city council. The programme detects visitors in the city, and creates a report of visitors split per day and day section of a given time frame. A day is split in sections of morning, afternoon, evening and night. Next to visitor counting a separate module classifies commuting in the city by analysing movement patterns and velocity of visitors.

This thesis is outlined as follows. In Chapter 2 the motivation for this research and related work is discussed. In Chapter 3 the research work is proposed and preliminary experiments are conducted to acquire requirements for the system and decide its design. Chapter 4 explains the main methods of the prototype tool created, and states the main problems in development of the tool. Following this Chapter 5 introduces experiments to validate the prototype and measure its performance. The results are listed in Chapter 6. The discussion is in Chapter 7. Finally the conclusion and future work is presented in Chapter 8.

2 Background

2.1 History

Over the past years urban growth is more and more accompanied with ICTs. The importance of ICTs is rapidly increasing in the last 25 years, enhancing the competitive profile of a city. New infrastructure for traffic management is connected to the city brains. Traffic, transport and accessibility is optimised using the information gathered by the city brains and used for further planning. Using ICTs for development is a key component of the smart city concept. However, the concept of smart cities itself is much larger than interconnected devices. ICTs in a city improve the availability and quality of knowledge communication and social infrastructure (Caragliu, Bo, & Nijkamp, 2011). The concept of smart cities introduce a strategic device for modern urban production factors. The term smart city is coined multiple times with a slightly different meaning. According to Hollands a key element in the literature is "utilisation of networked infrastructures to improve economic and political efficiency and enable social, cultural and urban development" (Hollands, 2008). This view focuses on networked infrastructure with smart needs. Although everything in a city might get connected, it does not imply a smart city. It needs to be activated using a smart application powering the brains of the city.

As ICTs were introduced in the 1990s and reached a wide audience in European countries, putting stress on the Internet as smart city identification no longer suffices (Caragliu et al., 2011). Instead a focus on smart economy, smart mobility, smart environment, smart people, smart living and smart governance should be used. These six topics are based on theories of regional competitiveness, transport and ICT economics, natural resources, human and social capital, quality of life and the participation of society members in the cities (Caragliu et al., 2011). Today, these cities represent a set of hyper-connected societies that enthusiastically embrace ICTs as key components of the infrastructure of modern cities.

2.2 Problems

In this chapter problems regarding data collection, inclusiveness and usability in context of smart cities are discussed. The main motivators for these problems are the vast amount of data available in a smart city, methods to process this information, and privacy of citizens observed by the systems.

Su, Li, and Fu (2011) state that smart cities depend on integration and release of massive urban spatial-temporal data and to obtain spatial-temporal data a breakthrough for more heterogeneous urban information needs needs to happen where high quality multi-source information is used. This leads to large-scale information and needs to be processed on a remote system. To facilitate these developments they argue that a sound information service with updated legal protection is needed.

One view in the literature is that "the smart city is all about systems that are connected to individuals who are plugged into digital information devices" (Calzada & Cobo, 2015). This does imply that every citizen is able to participate in the smart city and doing so enhances the quality of life in the city. However, "the existence of socio-technical systems, practices and strategies produce urban forms which intensify social fragmentation" (Puel & Fernandez, 2012). Thus, new technical local infrastructure affects communitarian life and should be considered before it is implemented in a large scale such as a city. The term smart city carries a positive and rather naïve stance towards urban development as Hollands (2008) noted.

Next to inclusion the availability and amount of information is important. Noted by Calzada and Cobo (2015) it is increasingly recognised that smart citizens have an interest in participating in a transition from controlled data mining to open access and user-centred systems. Information overload is increasingly common in a hyper-connected society. It is challenging to provide relevant information for improved decision-making without overloading citizens with endless data streams.

Depending on the application different solutions can be adapted and implemented to optimise feedback of information to crowds. In the next section some applications will be highlighted.

2.3 Adoption and applications

Although there might be a negative connotation as illustrated with the current problems, cities are adopting smart strategies to inform and help citizens. For example new cities in China are adopting the smart city paradigm by utilising smart transportation with smart traffic management systems enabling adaptive traffic signal control, smart public services, smart urban management and smart tourism to forecast tourism and promote development of tourism (Su et al., 2011).

In this chapter crowd management applications are discussed. Main focus of the applications is safety of large groups such as during festivals. Crowd management aims to guide large groups of visitors in the city and ease their stay. One example of crowd management is safety and enjoyability during events. In Zurich researchers deployed a mobile application to track visitors to create a model of the crowd density and movement during the largest Swiss event Züri Fäscht. Using incentives they managed to spread the crowd over the festival area and creating less high density spots. One of the incentives was a game guiding you around the festival. The game let you visit all places, but gave more incentives for moving to places that were less crowded. This approach as also effective for the larger acts. As often the crowd entered the area via the main points, those became hot spots for the crowd. Showing visitors that the other field looking at the same act had less people was a informational guide to move people there instead. Blanke et al. (2014) state that a game provides direct incentive to users to adopt smart sensing and participatory localisation as it provides direct benefits to the user.

Safety is one of the key concepts of crowd management. During large scale events with thousands of visitors a risk of stampedes is immanent. Smart crowd management could help managing and minimising these risks for events. Blanke et al. (2014) discuss the need for a careful design of the event area and schedule. Both are critical factors in crowd dynamics as the dictate the movement options during the event. One explicit suggestion they give is plenty of exit options such that the crowd after finishing the event can simply dissolve via multiple routes. Using information of previous events, event organisers should be able to get a head start in the critical process of organising the large group. The effects of attractions should be carefully planned and scheduled such that the crowd has no intention to move to certain points at exactly the same moment in time.

2.4 Techniques

Powering these applications are techniques developed for group tracking and local motion processing. The application of safety during festivals and crowd steering based on incentives operates on data mining using locally gathered position information in mobile applications (Blanke et al., 2014). The information gives insight in the most popular areas and the crowd movement between stage events. A stacked area graph visualising the movement is plotted in Figure 1. Using precise positioning a mobility graph with area density is created in Figure 2. One key element in this process of data collection is the participatory sensing, users consent with the retrieval and processing of localisation data in order to improve the event and gain small rewards.

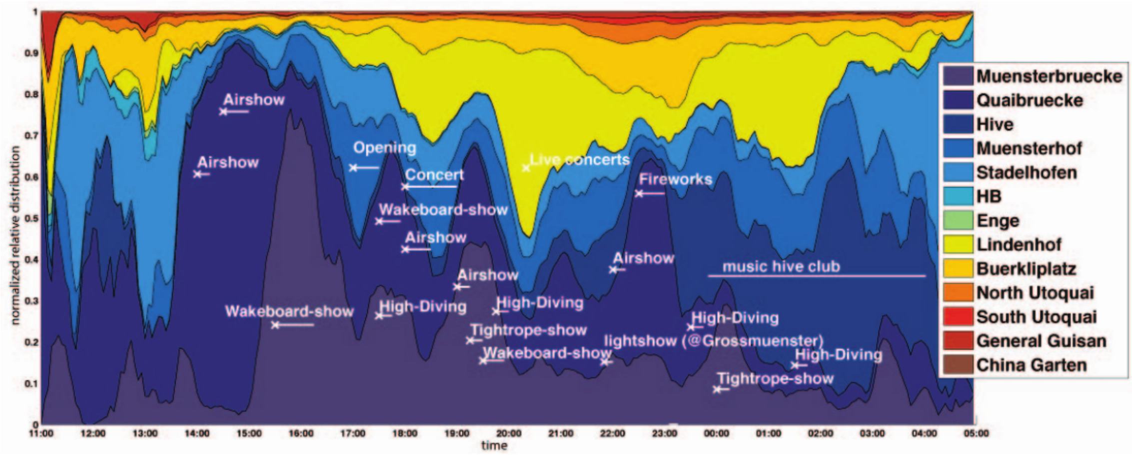


Figure 1: Relative visitor ratio between main locations. Adapted from (Blanke et al., 2014).

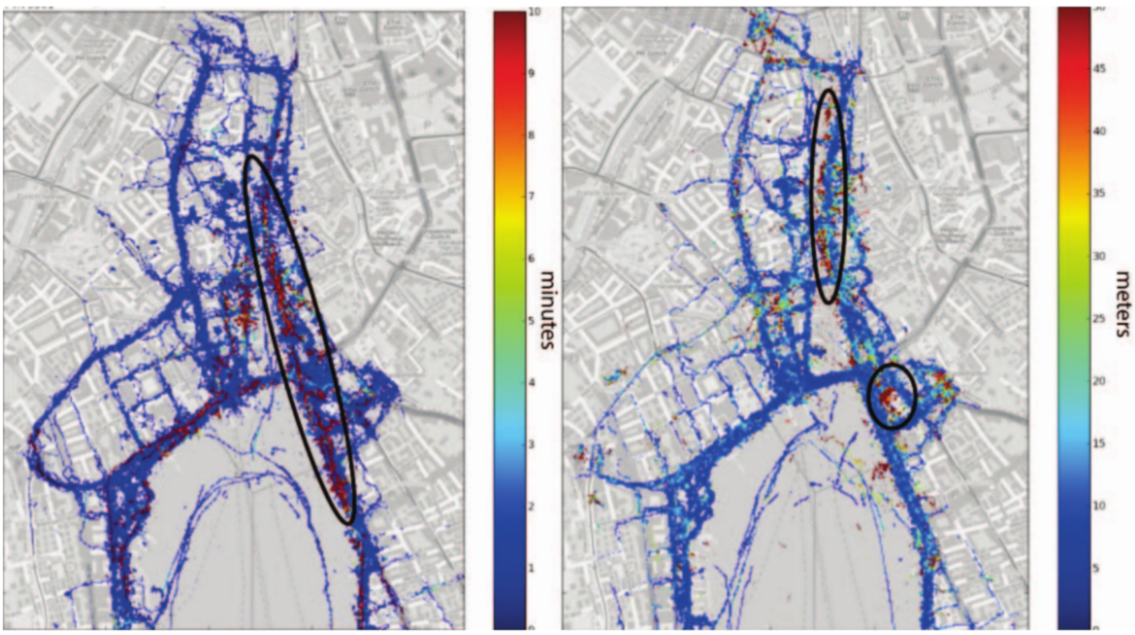


Figure 2: Delay of messages (left). GPS-accuracy from mobile devices (right). Adapted from (Blanke et al., 2014).

Continuing on crowd monitoring during festivals, Mallah, Carrino, Khaled, and Mugellini (2015) present group detection using smartphones. They state that vision based techniques are well used in event monitoring, with strategic camera placement and live feed to security crews. A major advantage of camera systems in the by-default all inclusiveness of all visitors, it does not rely on the collaboration of the crowd. However, difficult lighting and obstacles may impair the systems effectiveness. The field of view dictates how much of the crowd can be seen and observed for safety.

Another problem with camera systems is the scalability. Gong, Loy, and Xiang (2011) demonstrate that current deployed systems with manual inspection are not scalable. This is due to the complex deployment of the system, needed training for personal, and manual judgement of all critical situations observed. Using computer aided systems may improve this by automatically detecting unusual patterns and warn security personal.

They distinct two categories for observing crowd scenes. The first category is structured, where movement direction is directionally coherent over time. For example they watch train stations where crowds move in direction over the platform. The second category is unstructured, where the motion of the crowd at any given location is multi-modal. The unstructured category introduces extra challenges in object tracking as severe inter-object occlusion, visual appearance ambiguity and complex interactions among objects are present. The latter category is the often the case for festivals, making it harder to follow a crowd. Added to this uncertainty machine learning systems operating on these feeds often produce false positives, generating many warnings for security to investigate. Gong et al. (2011) argue that human assisted learning may reduce the number of false positives in such cases.

The smartphone based systems proposed by Mallah et al. (2015) is an alternative technology to monitor crowds to profit from the sensors embedded in smartphones. Like Blanke et al. (2014) the GPS location is collected from all phones, removing dependency on lighting and object occlusion for tracking and supports scalability for events. On the other hand, the on-device gathering of location required consent from the user. Furthermore, the high energy consumption during localisation and need for active network connection limit the usage of this approach to crowd monitoring. Their research mainly focusses on group detection and matches people to small groups.

Mallah et al. (2015) use crowd pressure as important metric of crowds. Crowd pressure is defined by Helbing, Johansson, and Zein Al-Abideen (2007) as dependency on local density (1) and local speed (2).

$$\rho(\boldsymbol{r}, t) = \frac{1}{\pi R^2} \sum_i \exp[-\frac{||\boldsymbol{r}_i(t) - \boldsymbol{r}||^2}{R^2}] \tag{1}$$

$$v(\boldsymbol{r}, t) = \frac{\sum_i v_i \exp[-\frac{||\boldsymbol{r}_i(t) - \boldsymbol{r}||^2}{R^2}]}{\sum_i \exp[-\frac{||\boldsymbol{r}_i(t) - \boldsymbol{r}||^2}{R^2}]} \tag{2}$$

Local measures are used over global ones, because human movement is different from liquid behaviour. Using these equations the collected data with unique identifiers for all users is transformed in crowd pressure information. Initial computations were quite lengthy with 5 minutes of computations for 500.000 simulated agents. For realtime applications some improvements are applied to reduce the dataset for computation. Only agents in a 2m radius around apoint of interest are included. The event area is clustered in sections of $1m^2$ to accelerate the location search. Instead of computing the local pressure for each agent, it is only computed for the centre of each $1m^2$ division. This reduces the execution time to 0.51s for 500.000 agents on 300x300m space.

Unfortunately, the collected data was not enough to provide statistical evidence relevant to group detection. The accuracy was high enough, but the weather conditions and small area of the event caused almost no movement during the event. Mallah et al. (2015) do show that when groups can be detected, evacuation plans can be tailored to groups and ensure everyone of the group receives the same evacuation plan and route to follow the same path. This solves the problem of splitting groups during evacuation by sending different routes to members of the same group. As groups have internal auto organisation, one person will lead the group and spread the information to all others. Members will proceed evacuating and take care of each other. Based on these observations Mallah et al. (2015) state that communicating an information to a group will be better perceived.

2.4.1 Fingerprinting

One common technique for localisation based on remote radios is fingerprinting. Fingerprinting is a state-of-the-art indoor positioning scheme currently widely deployed on various systems. It is radio technology independent as it combines all available information for a unique fingerprint of a location. Smartphones use it for indoor localisation using WiFi. A fingerprint refers tot he pattern of radio signal strength measurements recorded at a given location in space. It consists of a vector of identifier information (such as cellular Cell-ID, WiFi router MAC or beacon advertisement) and a corresponding vector of received signal strength values.

A typical situation is Android using fingerprints for indoor WiFi positioning to accelerate a global position fix. Faragher and Harle (2014) argue that movement through a complex signal environment, such as a building in a metropolian environment full of walls and objects, the received signal strength of any non-line-of-sight signal can vary rapidly on a fine spatial scale (sub metre level) as that signal penetrates different media. Fingerprinting operates on the principle that the received signal varies rapidly on the spatial scale, but very slowly in the temporal scale. Measuring the same position over time should record the same measurements within limits of measurement noise. The unique combination and fingerprint determines the location. In practise fingerprints inevitable degrade over time. This may be due to environmental changes as density of people within the building, position of furniture and even positions of walls and partitions. It is vital to update fingerprints and do regular resurveys to keep the database with fingerprints accurate.

Although most research focus on fingerprints in indoor situations, these are mostly chosen based on the challenging environment of rapid changing spatial radio reflections. In outdoor areas with typically more line-of-sight connections the richness of fingerprints increases. This makes it an interesting technology for massive outdoor tracking based on measured radio patterns by a mobile phone. Although finger printing is aimed at infrastructure independent positioning, observation using infrastructure may use these techniques to collect smartphone information and follow passing citizens.

To use fingerprinting a database containing all fingerprints is nessesary. Creating this database typically consists of two phases, an offline and online phase (Verbree et al., 2013). The offline phases builds the database by recording fingerprints in a radio map and by that way obtaining unique signatures of signal strength at various locations in the target area. The online phase compares the received signal strength of the radio with the earlier created radio map and give approximate locations based on the fingerprint.

2.5 Technologies

This chapter focuses on technologies to facilitate crowd management and localisation applications. First an overview of generally applicable techniques for localisation is presented. After this overview a closer look at WiFi and Bluetooth for localisation shows key concepts to the techniques and strengths plus weaknesses for the respective method. These two technologies have been picked due to the low energy requirement, ubiquitous deployment in modern urban environments and scalability in new areas. Both options also provide location from user device (eg. mobile phones) and infrastructure perspective (eg. equipment facilitating modern communication).

2.5.1 Trace data

Pan et al. (2013) discuss trace data as a source of smart city data. These traces provide important information on the mobility of moving objects (read humans van vehicles). Traces are becoming easily available as localisation technologies embedded in those objects are connected. A trace, generated by such a location technology, usually describes a temporal sequence of spatial points with corresponding timestamps. It is a simple input, but conveys underlying information on people and cities. For example crowds, traffic, human activity and social events. One common method for processing traces is mining. Mining can extract and reveal inherent information or knowledge about a city and its people. It enables the applications of a smart city and powers smart decisions.

Collection of trace data depends on the source of information. Sensors and devices could be used to detect location information, and report to a central system. This operation is "passive" as it requires not change or interaction with the to be traced object. Pan et al. (2013) divide trace sources in four categories:

- Mobile devices
- Vehicles
- Smart cards
- Floating sensors

Mobile devices such as phones and tablets are ubiquitous devices carried by humans. This group of portable devices are able to sent location information with the help of GPS, WiFi, GSM and Bluetooth. As the devices are owned by someone and carried along, the location usually mirrors the location of their owner.

Nowadays **vehicles** are more and more equipped with GPS devices for navigation services. GPS traces of a vehicle may not only depict the trace of the vehicle itself, but also that of its driver and passengers. Modern entertainment systems and navigation services include always-on options for software updates and live information.

Smart cards is a category of card used for transactions and authorisation. These cards typically interact with fixed location systems in the city. For example bank cards are used at fixed payment terminals and ATMs. Transportation cards might be a bit different. The swiping machines could be fixed on stations or floating through the city when mounted inside transportation vehicles. The exact location of the swiping machines is still known and all transactions could be tagged with location information.

Floating sensors is the last category. These are objects with localisation modules and report traces of itself. This is used in applications with object tracking, such as cargo containers. In smart cities postal companies may track cargo in transit to ensure timely delivery and improve delivery rates by optimising routes throughout the city.

An overview of all techniques and corresponding characteristics is displayed in Table 1. Often the more fine accuracy is paired with higher energy consumption. For applications this could be a parameter to tune depending on the required accuracy and power usage. In case of the festival monitoring a fine accuracy is required to detect groups and mobility of the crowd. As the area itself is relatively small, the detection only works with detailed information. For that purpose the researchers have opted to use GPS information. In other cases such as road occupancy and town square observation coarse methods may suffice.

An interesting detail of the technologies listed in Table 1 is the first actor receiving the location information. GPS systems provide accurate location information, but is only available to the receiver device. The signals are one way, and thus the user of the mobile phone chooses to share the GPS position.

Other systems like WiFi and GSM could operate in both directions. One option is from the device itself - by fingerprinting all radios and doing a lookup on the infrastructure location. Another option is localisation inside the infrastructure. In the last case devices do not have to participate in the infrastructure. Leaving traces such as scanning beacons gives WiFi infrastructure enough data to pinpoint devices based on the signal strength and location points where the signals were intercepted. This process is called WiFi sniffing. One major benefit of this method is all WiFi devices can participate and the localisation does not need involvement of the user. This technique is popular with in-shop tracking. WiFi devices constantly broadcast scan messages to observe whether a known station is nearby and connectible. Sniffing these signals allows tracking infrastructure to follow customers and determine the most popular routes through the shop.

Table 1: Comparison of popular localisation technologies. Adapted from Pan et al. (2013)

Technology	Data	Reference	Expression	Accuracy	Coverage
GPS	Geographic coordinate	Absolute	Physical	1–5 meters (95–99%)	Outdoors
WiFi	Access point ID + signal strength or local coordinate	Relative	Symbolic/ physical	1–20 meters	<100 meters from an access point
Cell Tower	Cell tower ID + signal strength or geographic coordinate	Relative/ absolute	Symbolic/ physical	50–200 meters in cities	Cell coverage. 5–30km from a cell tower.
Bluetooth	Device ID	Relative	Symbolic	Sensing range of Bluetooth	5–10 meters for Class 1; 20–30 meters for Class 2
RFID	Reader's ID/position	Relative/ absolute	Symbolic/ physical	Sensing range of RFID	1 meters for passive RFID; 100 meters for active RFID

2.5.2 WiFi based

WiFi based following give a high resolution image of movement. Sapiezynski et al. (2015) discuss that tracking human mobility with WiFi only need a few routers to create a strong connection with WiFi beacons. Using a small experiment one of the authors shows that in a time span of 48 hours over 3800 unique routers were recorded. Only a small number (8) are required to show 90% of the mobility during that time frame.

One of the authors demonstrates that in a timespan of 48 hour 8 access points visualise his daily pattern. The number one router is his home access point. The total time connected to this router is much higher than any other router. Looking at the pattern a block of continues connection is shown at night, where a smartphone remains connected to the infrastructure during sleep. The second highest connection time is the work infrastructure. Some less frequent connected routers placed in route and may be picked up as by-passers enter its broadcast range. For example during grocery shopping his phone connects to the public wifi available in the shopping centre. Over a time period of 48 hours these access points tell information on leaving home times, entering work, visiting the shopping centre, duration of shopping and returning to home time. A graph plotting the connected time is listed in Figure 3.

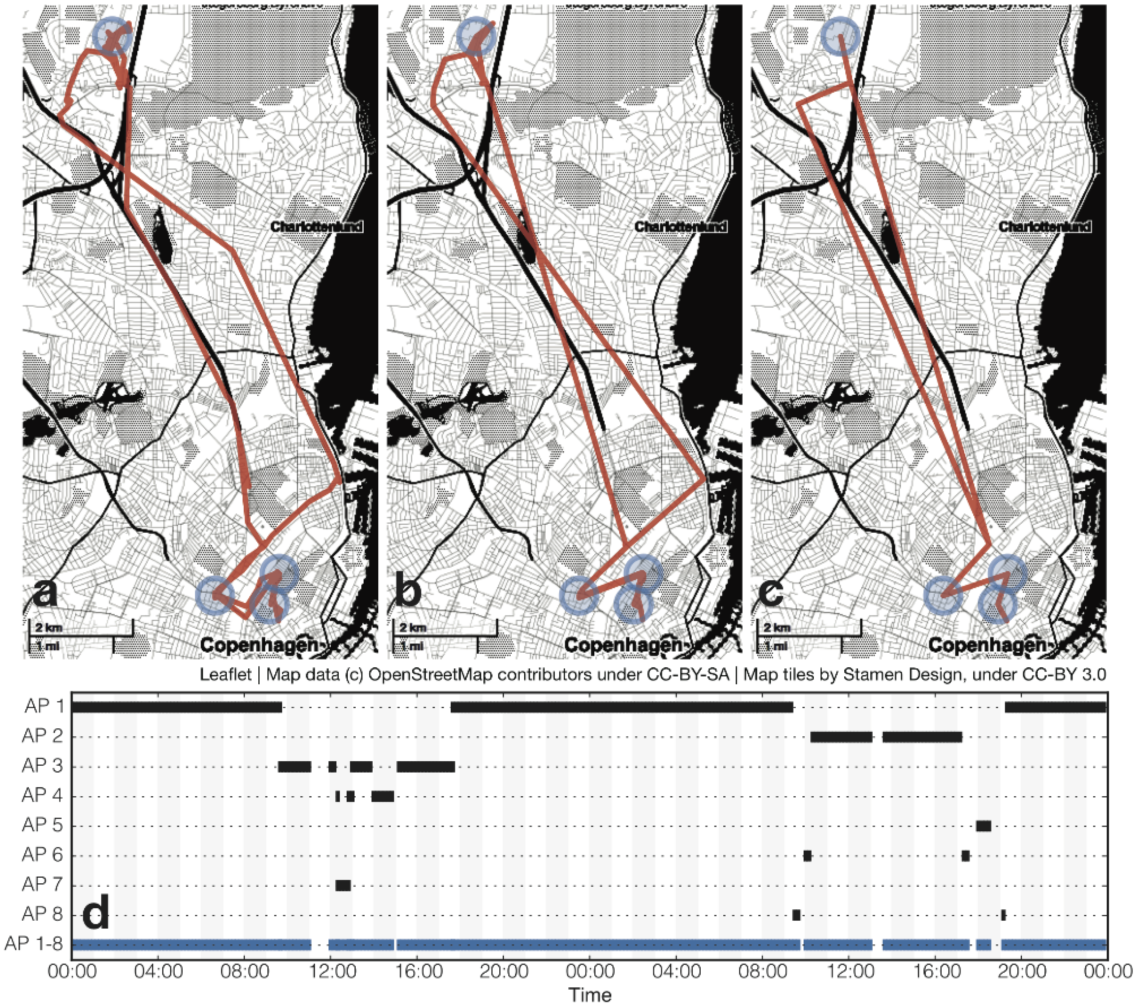


Figure 3: 48 hours of location data of one of the authors, with the four visited locations visited marked in blue: home, two offices, and a food market. Adapted from (Sapiezynski et al., 2015).

Using the fact that only a few Access Points, Verbree et al. (2013) tried infrastructure based user localisation with two WiFi monitors in the Hubei Provincial Museum. The infrastructure based method still requires an offline and online phase to train like the fingerprint method. This method creates a database containing X,Y and RSSI values resembling the radio map.

WiFi monitoring fingerprinting stems from the fact that no additional application is required to scan the APs and measure the RSSI levels.

To summarise infrastructure based WiFi monitoring:

1. Main advantage stems from the fact that no additional application is required to scan the APs and measure the RSSI levels followed by comparison in radio map on device.
2. All data is directly stored in the database of the organisation. The data is accessible

every moment of the day and can provide real-time information to the organisation for crowd analysis and density measurements.

3. It is possible to provide information to users about the location and crowded exhibits, this could be presented on large screens to include visitors without mobile phone.
4. The interval of monitoring scans is relatively big. Although visitors of the museum walk slow, this does affect localisation.
5. The received signal strength is reported differently in monitoring appliances than smart phones, a conversion is needed if the radio map is to be shared between both methods.
6. Storing the information in a database introduces privacy related problems.

2.5.3 Beacons (Bluetooth)

Another method for localisation is the use of Bluetooth Beacons. Introduced by Apple, iBeacon is a system of broadcasting nodes. Each node emits an identifier which is known to be at an exact location. In some systems this beacon could be mobile as well, using other stationary objects for a local reference.

One of the main advantages of Bluetooth is the lower energy consumption with comparison to WiFi. Bluetooth beacons use the Bluetooth Low Energy (BLE) standard. The range of the low energy variant is much more limited than WiFi communication. Although this may limit data transfer methods, it provides unique spatial fine-grained positioning as more beacons are required to cover an area. This does increase the proximity detection accuracy.

A common application is proximity advertising and information distribution. For example, shops may use iBeacons placed next to products to provide information on what a customer is looking at. Due to the low range, this works by proximity of up to a metre and may link customers to online content in addition to the physical product on display.

More interesting for tracking and localisation is using beacons for indoor positioning. Faragher and Harle (2014) have investigated the effectiveness of Bluetooth beacons for "indoor" positioning. In this context indoor means a high fidelity of beacon deployment and building structures interfering with the received signal. The BLE protocol defines three channels for broadcast advertisements, all nodes must broadcast on all of these three channels. These channels are labelled 378,38 and 39 and are centred on 2402 MHz, 2426 MHz and 2480 MHz. This is important as it spreads the signal to make it more robust versus interference on a part of the spectrum. It does influence the received signal as all channels operate in different areas of the 2.4 GHz spectrum, and may affect positioning accuracy.

In Figure 4 the deep multipath fades observed during a received signal strenght test using a beacon and access points is visualised. Faragher and Harle (2014) measured over 30 dB drops in power across just 10 cm of movement, and show that different channels exhibit fades at different spatial positions. The exact distances travelled by reflected signals is dependent on the wavelength of the signal, and fades occur at different positions for the different advertising channels. For WiFi the fades are notably less severe.

For fingerprinting this means that the fingerprint can vary dramatically over a short spatial range, even smaller than the expected accuracy of the system. This is amplified if

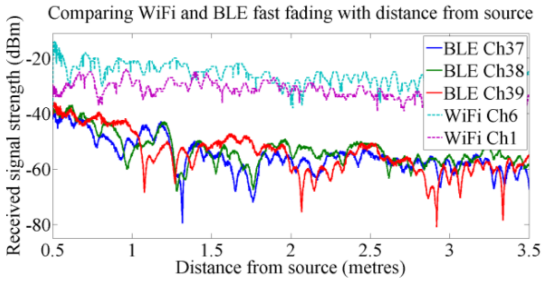


Figure 4: RSS variation over both time and distance for a phone moving on a slow conveyor belt away from a BLE beacon and two WiFi Access Points. Movement was a continuous millimetre per second to avoid Doppler effects. Adapted from (Faragher & Harle, 2014).

the receiver does not report the channel of the received advertisement and combines all channels as one input stream with different RSS values. Even in static environments this leads to changing fingerprints. Understanding and dealing with large fluctuations is key to producing accurate BLE positioning.

To summarise BLE for positioning:

1. Low bandwidth of BLE introduces more fast fading effects, and large RSS shifts. The use of three advertising channels by a BLE beacon, combined with frequency-dependent fading, can result in RSS measurements varying across a much wider range than WiFi.
2. Smoothing BLE RSS measurements by batch filtering multiple measurements per fingerprint is necessary to account for the bandwidth and channel hopping issues. The batch window is determined by the user velocity. The system shows best performance with a batch across a metre of user motion. Typically this means a window of 0.5 to 1 second in length.
3. Positioning accuracy increases with the number of unique beacons per fingerprint. Up to around 6-8 beacons display improvement of positioning accuracy, beyond this no significant improvement is seen.
4. Try to avoid the WiFi radio in a smartphone during BLE fingerprinting. There is some evidence to suggest thaat active WiFi scanning and WiFi network access can cause errors in the BLE signal strength measurements.

2.6 Summary

Smart cities are an upcoming phenomenon with smart paradigms to improve quality of life in a city. Building on different applications systems are integrated and citizens are informed of local mobility information to guide and aid in daily commuters patterns.

WiFi and Bluetooth both provide a good platform for localisation purposes. Depending on the application they offer different key strengths such as low energy for Bluetooth, larger coverage for WiFi, inclusiveness of visitors as smart phones are equipped with both radios, and fast positioning for crowd management applications.

For crowd based management the difference in infrastructure based tracking versus on device localisation can be neglected. In both cases the user needs to consent with providing information to the system and may optionally run the application. It does differentiate in ease of access as infrastructure based tracking requires no additional actions of an user but consent. This means larger groups could be included due to ease of participation. It also removes the need for local processing, which results in battery drains during festivals as connectivity is impaired due to huge quantities of devices connecting to the local infrastructure.

3 WiFi Dataset in Centre of City Enschede

This chapter explains the rationale of creating a prototype system to analyse the dataset. In the city of Enschede a public WiFi network is available covering the major squares in the centre of the city and adjacent shops. Everyday hundreds of visitors connect to the network as they visit the city. Such a large public network captures much of the city dynamics, as movement of shoppers and tourists pass through the area of coverage. The commercial entity exploiting the network is NDIX, a company providing services for broadband connectivity.

As part of this research, NDIX delivers a dataset containing information of sessions with all access points in the city. This dataset will be the main input to answer research questions regarding the city visitors and behaviour visible in the city. To analyse the dataset, a specially tailored tool will be created, operating on the dataset. This prototype tool is specifically for Enschede as it only operates on the NDIX set. The purpose of the tool is to automatically detect and map results computed on the dataset to questions open by the municipality regarding city dynamics.

3.1 NDIX network

NDIX is a platform for broadband infrastructure and IT services in the Netherlands. NDIX is the commercial partner of the municipality Enschede, operating and maintaining the public WiFi infrastructure in the city centre of Enschede. Since a couple of years the city centre of Enschede has a public and free available WiFi network. The main goal of this network is to facilitate broadband internet to all visitors of the city. Therefore the target is to maximize coverage of the city centre. As of today, the major squares in the city centre have good coverage and streets between the squares are mostly covered.

The WiFi network started in 2012 as promotion of Serious Request. Serious Request is a Dutch event supporting charity by raising funds during a one week period. In this week a large national radio station houses on a major city square to promote the event and raise funds for charity. This often attracts many visitors to the city hosting the yearly event. In the first year a few hotspots were set-up near the glass house to facilitate visitors of the event. This exploratory network was able to support roughly a thousand users. Over a couple of years this network expanded from a few points to full coverage from the Old Market to Van Heekplein, two main squares of the city centre. In 2017 the NDIX network had 11 nodes placed with 10 fully operational. Node #5 is temporarily removed as the building hosting the node is renovated. This node is planned to be returned on a nearby building. A map of nodes is shown in Figure 5.

The configuration displayed on the map in Figure is the configuration used in this research. Due to the missing node #5, the two squares are temporarily disconnected. All current network nodes are listed in Table 2 with their respective location. All nodes have a range of up to a few hundred metres. For example node #8 on the Van Heekplein stretches the full square. Nodes #1 - #8 are large outdoor nodes. Nodes #21 and #22 are indoor nodes, covering the shops in the Klanderij. Nodes #53 and #54 are directed nodes on the bus stop south of the city centre. These nodes have large coverage of the street with the bus stops.

The access points are outdoor Xirrus nodes, each containing 8 radios for a full 360 degrees coverage. The nodes support a large number of concurrent users, up to a few thousand per node. This ensures the network is capable of providing WiFi services during events. The deployment on various squares and shopping centres are located around popular regions of the city centre. For administration NDIX runs the Xirrus management software. This displays all current active sessions and stores all information in a database. The database is the primary source for WiFi application data in this thesis work.

As mentioned, the outdoor versions of access points have multiple internal radios. In an ideal scenario the active connection between a radio and smartphone might tell something about the relative location of the connecting person. Unfortunately, due to reflections, number of connections and other factors affecting wireless transmissions it might be the case that opposite radio offer a better link and is the preferred connection. This limits the value of knowing the radio with respect to locating a person. It does introduce a interesting detail, roaming devices on a square may start new sessions on the same location. This may differentiates mobile devices and stationary devices.

As the WiFi network is publicly available, and provides large coverage of the city, the local council is interested to learn whether this configuration can assist in counting and monitoring visitors. The main application would be traffic observation, safety of large crowds moving throughout the city and observation of city activities. This new application of the WiFi network is an experimental research towards the smart city framework.

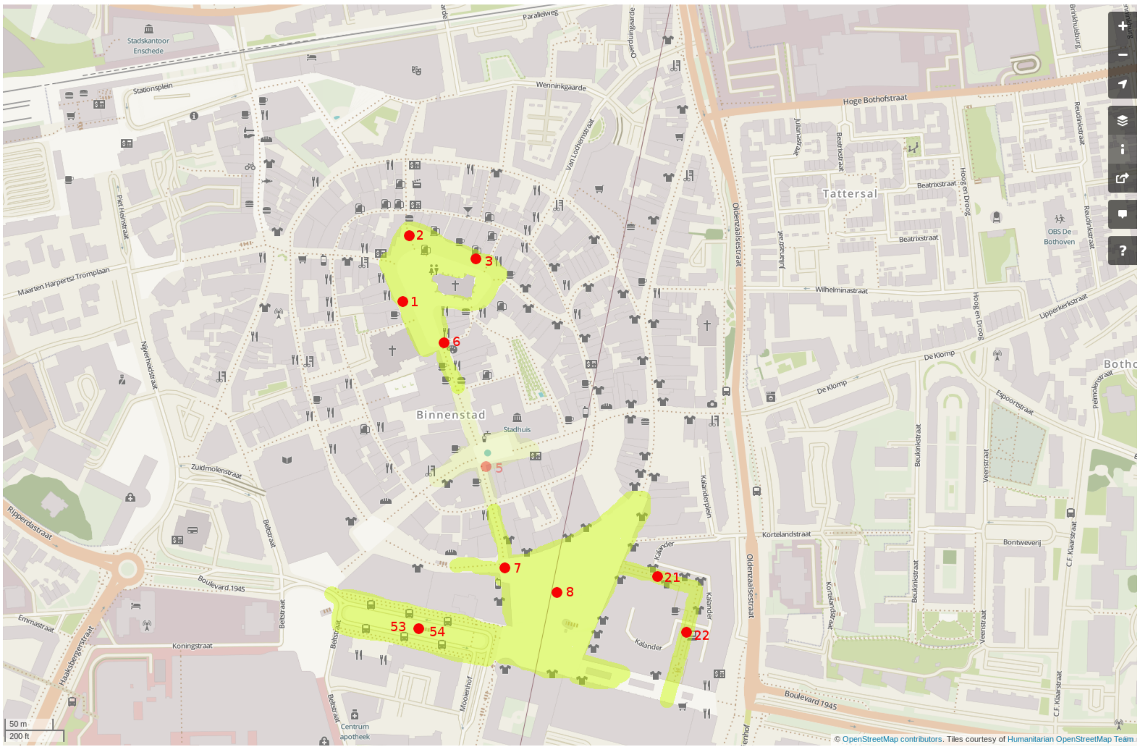


Figure 5: Map displaying all network nodes of the NDIX WiFi network in Enschede in 2017. Node #5 is dotted as transparent, this node is not active.

Table 2: Table listing all NDIX WiFi nodes in the city centre.

Node	Network	Location
AP-ENS-01	Eduroam, Enschede_Stad_Van_Nu	Latitude: 52.220489, Longitude: 6.895032
AP-ENS-02	Eduroam, Enschede_Stad_Van_Nu	Latitude: 52.221131, Longitude: 6.895705
AP-ENS-03	Eduroam, Enschede_Stad_Van_Nu	Latitude: 52.220841, Longitude: 6.896646
AP-ENS-05	—	Latitude: 52.219241, Longitude: 6.896084
AP-ENS-06	Eduroam, Enschede_Stad_Van_Nu	Latitude: 52.220203, Longitude: 6.895788
AP-ENS-07	Eduroam, Enschede_Stad_Van_Nu	Latitude: 52.218229, Longitude: 6.896529
AP-ENS-08	Eduroam, Enschede_Stad_Van_Nu	Latitude: 52.217598, Longitude: 6.897556
AP-ENS-21	Eduroam, Enschede_Stad_Van_Nu	Latitude: 52.2180544, Longitude: 6.8989162
AP-ENS-22	Eduroam, Enschede_Stad_Van_Nu	Latitude: 52.217574, Longitude: 6.899294
AP-ENS-53	Eduroam, Enschede_Stad_Van_Nu	Latitude: 52.21742, Longitude: 6.895350
AP-ENS-54	Eduroam, Enschede_Stad_Van_Nu	Latitude: 52.21742, Longitude: 6.895350

3.2 NDIX WiFi dataset

NDIX provides a dataset with session records of its WiFi network for scientific research. This set is an export of the local management database, and contains all records as stored by NDIX. All records altered for privacy obligations, no personal identification is stored in the dataset. The records are anonymized by removing the MAC addresses of all sessions, and replacing it with unique increasing integer identifiers. Repeated visits within a defined timespan are assigned the same MAC address replacement.

Therefore, visits with multiple sessions are still linkable. Effectively, all records for person with MAC X are now $ID(1)$ and MAC Y is $ID(2)$ for all sessions following within a month. When person X visits again after 30 days, identifier $ID(3)$ may be assigned. The dataset itself contains a time span of July 2015 to May 2016. This means no recurring months are available, limiting comparison between years in the city.

As the set must be anonymized prior to release for scientific research, manual conversion is needed for release of a new dataset. The latest available information is the set with sessions up to May 2016. To aid with research and pattern recognition in experiments, NDIX provides device specific traces which are not anonymized. NDIX exports traces related to devices which will be used in experiments. This information is useful for mapping system traces with defined behaviour in experiments, as each device can be analysed individually.

The format of the records in the dataset is as follows. Each record has a unique identifier, the replacement for MAC. Next to the identifier is the connected access point, timestamps of the session and signal strength. In Table 3 the record is explained in detail.

The following example illustrate the records stored in the dataset. All records originate from access points and describe one particular session of a connected device.

```
"1","AP-ENS-01",1,1461439705,1461439845,140,"iap3","eduroam",40,-80
"2","AP-ENS-02",1,1461439845,1461439865,20,"iap1","eduroam",40,-70
```

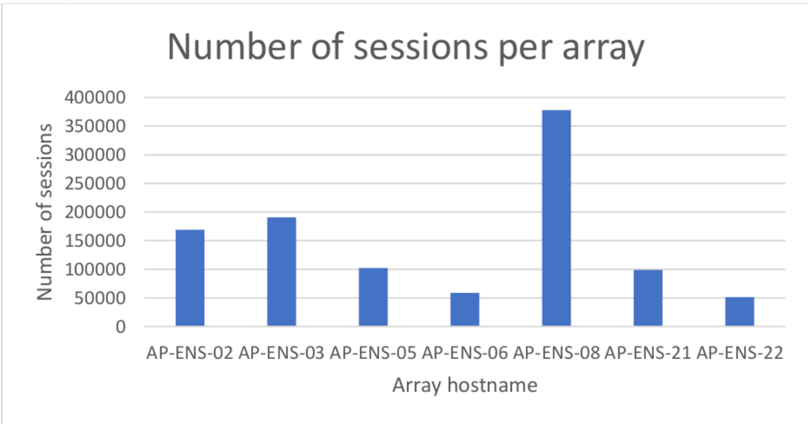
Table 3: Data fields recorded in each session using the NDIX WiFi network

Column	Metric	Description
1	ID	Unique identifier of the session
2	Arrayhost	The internal name of the connected access point
3	MAC	MAC of the connected device
4	session_start	Start time in unix timestamp format
5	session_end	End time in unix timestamp format
6	session_length	Duration in seconds
7	IAP	Internal radio
8	Network	Name of the connected network
9	Channel	Channel ID used in this session
10	RSSI	Average RSSI of the session

Some interesting observation of the dataset is that session_length may be negative. For example in October when daylight savings is enabled, the clock will be reversed by one hour. In this particular event the session could be negative by one hour. The dataset contains multiple records with negative lengths, mostly caused by DST.

3.3 Dataset discovery

Designing a method to retrieve the information in the dataset to answer the questions it is important to inspect the dataset and analyse the basic structure. To get a feeling of the dataset and its contents the set is explored and evaluated. The dataset of NDIX is imported in PostgreSQL. This advanced database system allows sophisticated queries on the dataset. Using counting, sorting and grouping queries basic information of the dataset is acquired. This reveals the timespan of July 2015 to May 2016. In this timeframe a total of $1.048.575$ sessions are recorded with 52.367 unique devices. The set shows three networks broadcasted on the nodes, Eduroam, Enschede_Stad_Van_Nu and VRT-OOV. VRT-OOV is only listed prior to any connections to Eduroam and Enschede_Stad_Van_Nu, likely a test network used during installation of the network.



array_hostname	count
AP-ENS-02	169014
AP-ENS-03	190445
AP-ENS-05	102201
AP-ENS-06	58917
AP-ENS-08	378074
AP-ENS-21	98852
AP-ENS-22	51072

Figure 6: Barchart showing all access points and total number of sessions

Number of sessions grouped by access point show that not all access points are included in the list. AP-ENS-01 and AP-ENS-07 are not recorded in the dataset. It is unknown why these access points are not included, as they operate live during the research period. NDIX responded that only node $\#5$ is currently offline due to hardware failure. Next to the missing nodes another outlier is AP-ENS-08, it is clearly the access point with most sessions. This access point is installed on the largest square in the city centre and provides full coverage of the square.

One interesting remark is that the dataset only contains long sessions. It is not rare to find sessions with multiple minutes of length. For tracking global movement throughout the city this is sufficient. However, for local motion and classifying mobility these sessions may be to lengthy as they hide all movement for that length of time frame. Another disadvantage of classifying mobility using this dataset is missing reference information. None of the records describe the mobility of the object, thus the system cannot learn using the dataset.

4 Methodology towards analysing city visitors patterns

This chapter describes methods defined to analyse the dataset. The goal of the analytic is to estimate the presence of visitors in the city and the mobility chosen for moving throughout the city. The first operation performed on the dataset is to reduce its size. By filtering the sessions and connecting dots the first step produces paths. Paths are a summery of the traversed route by someone in the city, concatenating all recorded events within a set time frame. The creation of paths is explained in section 4.1. Based on the created paths, methods will provide counts for the number of visitors in the city for a given time period, detect events with large number of visitors joining, find shopping behaviour in the dataset and finally, mobility mode. Paths are the basic input block for all methods developed.

A prototype programme to read and analyse the dataset is developed in this work. The programme is partly written in Golang and partly in Python. Golang is a programming language developed by Google introducing a new concept of inter process communication via message channels. Additionally, Python helps analyses on the final results due to the need for statistical support libraries such as numpy, pandas and seaborn. These tools help detecting events using moving averages, standard deviation of the set and plotting trends in the results.

For initial testing and validation of the system a reduced version of the dataset is used. This reduced set only contains the month April. April is interesting as it offers multiple events spread throughout the month. It has multiple market days, weekend with varying population counts and Kingsday. April could be compared to the same time period in 2017 as this is in the period of this work. By comparing the same month in multiple years the system could determine growth of the network and or traffic in the city. It will reflect changes over longer period of time.

For classification of mobility the system needs labelled reference input. To obtain ground truth input, experiments will be conducted in the city centre, creating traces of mobility patterns. To create a dataset NDIX needs to consult an external company for extraction of the data and anonimisation. This is a lengthy process and needs explicit funding in order to proceed. For a small number of devices, NDIX is able to quickly export traces containing a device MAC, connected hotspot and timestamp. As traces differ in metadata compared to dataset records, traces need to be converted into records for a unified processing method. The prototype should not differentiate between the dataset and additional traces. Using these traces the system should learn patterns and apply these to the large dataset.

4.1 Path creation

The dataset listing the WiFi sessions contains session records of all connected devices to the network. These session records have an identifier, device address, start timestamp, end timestamp, access point name and signal strength. Path creation is the process of converting the dataset in small chunks to work on using the analyser.

The modelled data is a layered abstraction building on these sessions (s). The timespan between two sessions represent mobility of a WiFi enabled device. Therefore, a segment (S) which connects two sessions shows the local mobility of a device and user. In a segment the source and destination access point are stored, including the interval time between the sessions. Using the segment's timespan, the distance travelled is computed. As the records do not contain a representative RSSI, mobility is based on the distance between access points, which is roughly the distance between the access points as all are located on street level. A map of access points contains the exact latitude and longitude of each access point. In addition with the interval time the mobility velocity is computed for each segment.

Paths (P) summarise the mobility of sessions, and feed the analytic system. A path represents a trial of a visitor in the city. A configurable parameter time (t) defines the maximum timespan between segments to be considered as one path. For example, a time period of 15 minutes connects all sessions recurring within this time frame of 15 minutes. If a user leaves at 14:30 and reappears at 14:40 this is considered as the same trip in the city centre. After this timeframe any new connection is considered as a new visit to the city. However, if a user connects at 14:30 and remains connected until 15:30 with the same access point it may have been a stop at a café. Any mobility following this session is still concatenated as one as the user remained connected to the network. The mobility of a path is averaged over all segments connected in that particular path.

In short, paths are built by concatenating segments, where the N is the length of the path. The system will store a set of paths, linking devices to all their paths travelled in the city. Segments (S) consist of exactly 2 sessions (s) and the interval between sessions for mobility displacement. A segment summarizes the gap between two consecutive sessions. This gap shows the local mobility in the path. Sessions (s) are the actual rows in the dataset. Sessions are the spatial representation of physical presence with a location. Segments and paths describe the temporal displacement in the system of access points. In Figure 13 a path with two segments and three sessions is illustrated.

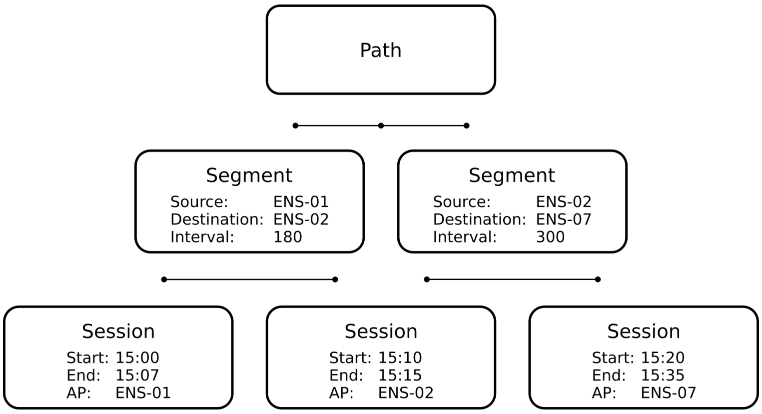


Figure 7: Example of a path with $S_N = 2$ and $s_N = 3$.

The algorithm for path creation and mapping to user devices is displayed in Listing 1.

Listing 1: Algorithm: path creation

```
input: List with sessions,
output: Map with mac addresses containing lists of paths

mac_count = {}
mac_last_pos = {}

mac_paths = {}

for session in sessions:
  mac = session.mac
  mac_count[mac] = mac_count.get(mac, 0)
```

```
  last_pos = mac_last_pos.get(mac, None)

  if last_pos:
    segment_element = mac_last_pos[mac]
    time_passed = session.start_time - segment_element.end_time
    segment = SegmentTime(session.id, segment_element.array_hostname, session.array_hostname,
    segment_element.end_time, time_passed)

    user_path = mac_paths.get(mac, None)
    if not user_path:
      mac_paths[mac] = [[]]

    if time_passed < maxTimeBetweenSegments:
      mac_paths[mac][-1].append(segment)
    else:
      mac_paths[mac].append([])

  mac_last_pos[mac] = session
return mac_paths
```

4.2 Counting devices

The counting component estimates the total number of devices per day in the city centre. A unique counter on MAC filters sessions to provide a rough estimate of visitors. The output of this component is a list of all months and days and corresponding number of devices in the city. This gives an estimate on the population throughout weeks and months. For example, it provides input for event detection which notices spikes in population numbers and flags that day as an potential event. The algorithm for counting is in Listing 2.

Listing 2: Algorithm: Path counting

```
        input: List with paths,
        output: Map with counts per month and day

month_day_split = {}
for p in paths:
  for x in paths[p]:
  try:
    year = x[0].start_time.year
    month = x[0].start_time.month
    day = x[0].start_time.day
    key_tuple = ("{}-{}-{}".format(year, month if month >= 10 else "0" + str(month),
     day if day >= 10 else "0" + str(day)))
    month_day_split[key_tuple] = month_day_split.get(key_tuple, 0) + 1
  except IndexError:
    pass

sorted_month_day_split = sorted(month_day_split.items(), key=operator.itemgetter(0))
return sorted_month_day_split
```

4.2.1 Counting visitor hotspots

The visitor counter extends the basic counter with day segmentation. Each day is split in four parts, morning, afternoon, evening and night. Paths are recreated and stored in the specific bucket of corresponding day segment. Using these four distinct time frames gives insight in the characteristics of the city. The population shifts area throughout the day. It is expected to see more visitors in the shopping area in the morning and afternoon, while the pubs should attract more visitors in the evening and night. The reduction from device to visitor is due to macs only add one path to the set. The algorithm for splitting the paths in four segments is listed in Listing 3.

Listing 3: Algorithm: visitor counting

```
input: Dataset lines with sessions for a specific time frame
output: Paths created based on sessions, split per day section

# Map storing the number of sessions per device
deviceSessions := map[string][int]

# Lookup table for the last session of a specific MAC
macLastPosition := map[string][NDIXRow])

# Maps for storing segment per edge per time of day
pathTraffic := map[string][[]SegmentTime])
pathTrafficMorning := map[string][[]SegmentTime])
pathTrafficAfternoon := map[string][[]SegmentTime])
pathTrafficEvening := map[string][[]SegmentTime])
pathTrafficNight := map[string][[]SegmentTime])

# Stores segments per hour of day
pathTimeSplit := make(map[int]int)


line := scanner.Text()
el := ParseNDIXLine(line)

# Count number of device sessions
deviceSessions[el.Mac] = m[el.Mac] + 1

# If a path exists (eg. there is a previous session), append and store gap
# time
if pathEl := macLastPos[el.Mac]:
  route := pathEl.ArrayHostname + "," + el.ArrayHostname
  timePassed := el.StartTime - pathEl.EndTime

  if timePassed < maxTimeBetweenSegments {
    segment := SegmentTime{ID: el.ID, ArraySource: pathEl.ArrayHostname, ArrayDest: el.ArrayHostname, StartTi
    pathTraffic[route] = append(pathTraffic[route], segment)

    # Add segment to correct time bucket
    tm := time.Unix(int64(pathEl.EndTime), 0)
    pathTimeSplit[tm.Hour()] = pathTimeSplit[tm.Hour()] + 1
    if tm.Hour() >= timeBeginMorning && tm.Hour() < timeBeginAfternoon:
```

```
      pathTrafficMorning[route] = append(pathTrafficMorning[route], segment)
    else if tm.Hour() >= timeBeginAfternoon && tm.Hour() < timeBeginEvening:
      pathTrafficAfternoon[route] = append(pathTrafficAfternoon[route], segment)
    else if tm.Hour() >= timeBeginEvening:
      pathTrafficEvening[route] = append(pathTrafficEvening[route], segment)
    else if tm.Hour() >= timeBeginNight && tm.Hour() < timeBeginMorning:
      pathTrafficNight[route] = append(pathTrafficNight[route], segment)

macLastPos[el.Mac] = el
```

4.3 Detecting events

The events detector uses the counting algorithm introduced with the path counter. It extends this algorithm by taking the population umbers per day as input and calculate a moving average over the full set. Using standard deviation and moving average significant outliers are detected and reported as potential events. Events must be statistically significant in order to report, small variations indicate no difference in attendance for events.

It is expected to see more significant outliers throughout the year and in smaller windows such as monthly. This due to large national holidays which attract visitors to the city. In smaller windows such as one month local events like market day may increase the attendance. In this case it is expected to see more visitors in Tuesday and Saturday. The input timespan is variable, it may range from a few days to the full dataset.

The detection of anomalies operates as follows. After generating the paths and counting visitors the event detector receives a list of all days and corresponding visitor counts. This list is sorted on month and day, the date of the count. The counts are extracted and stored as a numpy array to use in the analysis. Numpy first computes the moving average and residual, the difference between a specific count and moving average. The variance in the residual is used as metric to detect outliers. The algorithm accepts σ as input to set the significance needed for an outlier. The output is a map with date and count listing all detected outliers. The algorithm for event detection is in Listing 4.

Listing 4: Algorithm: event detection

```
input: y: List of path counting, window_size: Moving average size, sigma: threshold for anomaly)
output: List of anomalies, date of occurance and number of paths counted

y_avg = moving_average(y, window_size)
residual = [y - avg for y, avg in zip(y, y_avg)]

# Calculate the variation in the distribution of the residual
std = std(residual)

anomalies = {}
for index, y_i, avg_i in zip(count(), y, y_avg):
  if y_i > avg_i + (sigma*std) or y_i < avg_i - (sigma*std):
    anomalies[index] = y_i

return anomalies
```

4.4 Detecting shopping behaviour

The shopping behaviour detector distinguishes visitors in two categories. The first category are visitors with a "goal", moving targeted to one shop, buying one product and returning back home. These visitors typically have a shorted stay in the city. The second category are window shoppers, people who traverse the city looking for new or interesting items. This category typically moves slower and spends more time in the city, and may visit multiple places.

Shopping behaviour is detected by looking at the number of visitors in the shopping area of the city. The main shopping areas in this system are the Klanderij and Van Heekplein. The corresponding access points for these areas are AP-ENS-07, AP-ENS-08, AP-ENS-21 and AP-ENS-22. AP-ENS-05 should be included as a link between the shopping are and the pubs, but is unfortunately not operation during the recording of the dataset.

The shopping detection looks at the duration of the path to roughly estimate the shopping behaviour. Short durations indicate targeted shopping, a visitor attends a shop for a particular product and leaves quickly after acquiring the product. Longer stays may indicate visitors window shopping and looking for new products or gifts. As single paths may not capture the full traversal in the city due to losing connection in shops, this algorithm looks at all paths for a given visitor on one day.

4.5 Commuting and transportation mode

The commuting detector tries to classify the paths and according metadata to find the method of transportation. The current iteration classifies pedestrian and bicycles. Due to the limitations of the system the classifier extracts additional metadata in records for classification. In one of the experiments the starttime of session 2 was before the endtime of session 1. Therefore, depending solely on the time span between sessions to detect movement between access points is not a valid observation. The time variance introduced by the system for leaving sessions and reconnecting to the previous access point is not consistent, some records stop at one minute after disconnecting from the access point, and other sessions after ten minutes. To compensate for this behaviour of the system a skipped node variable is introduced. High velocity transportation, eg. bicycles, may skip occasionally nodes ad the time to connect is longer than the airtime of the node.

Therefore, important metadata include the time traversed on the path between nodes and whether nodes have been skipped. If a node in a path between squares is skipped in a short timespan it is likely to be a bicycle taking an alternative route than the routes covered with the WiFi network. Another option could be fast movement, resulting in such fast passing of an access point that the time required to establish a WiFi connection is longer than the time spend near the access point.

As the locations of all access points are known, the system computes the travelled velocity based on the node distance of access points and gap of the segment. The system needs two training sets, one for learning and adjusting the points for metrics and a secondary set for testing. Using only one set may result in overfitting where the output is exactly matching the input of the set. Keeping the sets separate measures the performance of the detection on future sets which will not be trained.

$$\sum_{S_i \in Path(I)} DecisionMobilityModel(P_i) \tag{1}$$

Actual classification is based on a decision tree with weighted features. Firstly the speed is considered. This metric is computed in metres per second using the node distance and time gap listed in the segment. The use of segments and not full paths for mobility detection is justified by the occurrence of circular routes. For example, if one is to enter the city via AP-ENS-08, walks up to AP-ENS-07 (Primark) and leaves via AP-ENS-08 the path starts and end on the same node. Computing the time difference and travelled distance is not possible, as there is no distance in the nodes. Using a summation of all segments a score for the full path is computed.

Listing 5: Algorithm: Commuting classifier

```
input: Dataset lines with sessions for a specific time frame
output: Paths created based on sessions, split per day section

# DecisionMobilityModel Uses start point, end point and time between
# to find type of movement
# If a node is skipped and time is low, it may be a bicycle
func DecisionMobilityModel(startNode string, endNode string, time int) float:

  # Compute score for walking vs bicycle
  var score = 0.0

  # Was a node skipped in the network?
  if IsNodeSkipped(startNode, endNode):
    score += weightTrainedSkip

  # Find average speed if available
  score += weightTrainedVelocity * compVelocity(startNode, endNode, time)
  return score

# Compute the rough avarage velocity based on node distance and time
func compVelocity(startNode string, endNode string, time int) float:
  # Find distance between node A and B
  distance := DistanceBetweenAP(NDIXAPs[startNode], NDIXAPs[endNode])
  if time < minimumSessionTime:
    return 0.0

  if distance > 0 && time > 0:
    return distance / float64(time)

  return 0.0

# Check whether the link between start and end skips a node
func IsNodeSkipped(startNode string, endNode string) bool:
  # Sort path
  if endNode < startNode:
    startNode, endNode = endNode, startNode
  switch:
  case startNode == "AP-ENS-01" && (endNode != "AP-ENS-02" && endNode != "AP-ENS-06"):
    return true // Skipped 06
  case startNode == "AP-ENS-08" && endNode == "AP-ENS-06":
    return true
  default:
```

```
    return false

func ComputeScore(p Path) float:
  var score float = 0

  for seg in path:
    decision := DecisionMobilityModel(seg.ArraySource, seg.ArrayDest, seg.Interval)
    score += decision
  return score
```

4.5.1 Walking

Typically walking is a slower movement throughout the city than biking. For walking the threshold score is set at less than 2.0. Walking may occasionally report false positives due to fast switching access points. This may be true in the event of walking on the maximum range of two access points and continuously switching between the access points. This algorithm only considers the distance between the access points, not the users distance. Therefore, the reported distance could be hundreds of metres in a short time span of a few seconds. To minimize this behaviour a minimum session length is defined prior to adding the session to the path distance travelled.

4.5.2 Biking

For biking a detection threshold of movement larger than 1 metre per second is set. In addition with the other parameters describing the route biking is classified with a score of more than 2.0. This value will be changed during a training phase.

5 Data collection and experiments

This chapter discusses the data collection and experiments conducted in this work. The main source for data analytics is the NDIX dataset containing wifi sessions. This set is used for counting visitors in the city, event detection based on anomalies and algorithm validation for walking and biking. As the NDIX dataset has no reference information, the mobility detection for walking and biking is only tested to see if the large amount of data can be processed using this algorithm.

In addition to the NDIX dataset are traces recorded via the NDIX network. These traces have context as they emulate visitor behaviour in the city in a predefined way. The exact scenarios are detailed in section 5.1 describing the test plan for mobility classification.

A third source for the experiments is recorded simultaneously with the traces in the NDIX network. Using a reference device, with long range WiFi antenna, a high density chain of network connections is recorded. This device stores one point per second which include WiFi RSSI and GPS location. This reference data is compared to the traces in order to see if enhanced WiFi data may improve the accuracy of mobility detection.

The following table lists all experiments and dataset input. T1-4 are traces recorded according to the test plan in section 5.1.

Table 4: Table showing each experiment and used dataset

Dataset	Counting	Events	Shopping	Walking	Biking
NDIX	X	X	X	-	-
T1: Walking				X	X
T2: Cycling				X	X
T3: Shopping			X	X	X
T4: City entrance			X	X	X
RPI: Reference				X	X

5.1 Test plan mobility classification

The main goal of this test plan is to record traces with context for mobility classification. During the recording of traces labels are noted for all mobility patterns and movement in the city. In this experiment multiple devices are active at once, creating an artificial group. Used devices are four smart phones and one tablet. This represents users with varying type of devices. For reference measurements a Raspberry Pi is included, equipped with long range antenna it records the signal strength of all visible networks. Next to the WiFi antenna the Raspberry Pi is fitted with a GPS module, reporting the GPS mixed with GLONASS location every second.

The Raspberry Pi runs a script in Golang to scan all WiFi networks every ten seconds, which is roughly the time it takes to perform a full scan. For each network the GPS location, network identifier, channel, link quality and noise level. Meanwhile it tries to connect to Eduroam and stays connected. It records the signal strength of Eduroam every second, which is the fastest recording rate offered by the Linux tools. The output of both measurements is stored in a logfile on the Raspberry Pi.

Figure 8: Overview of all used devices. From left to right: iPad2, HTC Desire, Samsung Galaxy S3, Samsung Galaxy S4 and Raspberry Pi with long range antenna on top. The Nexus 6P is missing in this picture, it shot the picture.

Table 5: Overview of devices in the experiment

Device	Network	Expected results
Nexus 6P	Eduroam	Main device, continuously visible via Eduroam.
Galaxy S4 mini	Eduroam	Baseline device as carried by visitors
Galaxy SIII	Enschede_Stad_van_Nu	Producing equal traces to the S4 but on the other network
HTC Desire	—	Not producing a trace as its not connected
Apple iPad2	Eduroam	iPad may not connect in standby and show few traces
Raspberry Pi	Both	Measures link quality using a long range antenna

Scenarios A total of four traces are recorded, labelled T1 to T4 in the scenarios. Trace T1 and T2 focus on mobility, by traversing the city as pedestrian and on a bike. T3 and T4 focus on shopping movements as a pedestrian.

The scenarios for this experiment are multiple walking and cycling activities. For each scenario the objective with route is listed and a hypothesis. Next to the scenario the route is plotted on a map to visualize the path. Scenarios are categorised in walking, cycling, shopping and entry-point evaluation.

The scenarios will be performed on two consecutive days, recorded in the afternoon. First the scenarios are played on a Tuesday. This is a market day in Enschede and includes more visitors on the Van Heekplein square. On Wednesday the scenarios are repeated. Between all scenarios the devices are moved out of coverage to reset the recording of sessions. Including the market in Enschede might reveal network performance changes on a busy market square. Many visitors will traverse the square and stop at various vendors. It is interesting to see if the market stands influence network connectivity and visiting guests.

The scenarios are played in the afternoon. As the scenarios are performed on two days the results can be compared. For training of the mobility detector the results of the first half of the afternoon of both days will be used as training set. The second half of the afternoon will be used to validate the training of the model.

T1: Walking

The first scenario is a walk from north to south. This scenario consists of two parts, one with the smart phone on standby and one wherein the smart phone is actively used. The first case emulates someone traversing the city walking towards a goal. The second case emulates someone who searches for a specific store or wants to look up information online. As smart phones may take up to a half hour before they enter standby mode the passive case is always conducted first.

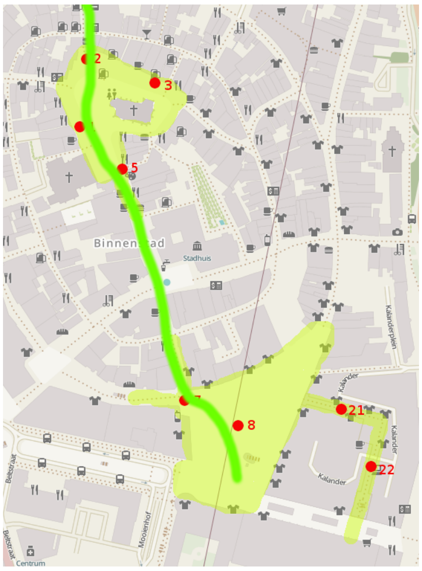


Figure 9: Route for the walking scenario

Passive use

Objective: Walk path through city from church to market via *(2) -> (1) -> (5) -> (7) -> (8)*

Hypothesis: Baseline for movement sets a continuous connected path. As the phone is in standby, the WiFi radio might be shutdown to conserve energy. This could disconnect the path at some nodes.

Active use

Objective: Walk path through city from church to market via *(2) -> (1) -> (5) -> (7) -> (8)*

Hypothesis: Baseline for movement sets a continuous connected path. As the phone is connected WiFi gaps are minimal, quick reconnects happen at all nodes.

T2: Cycling
Objective: Cycle path through city from church to market and back via *(3) -> (2) -> (1) -> (5) -> (7) -> (8) -> (3)*
Hypothesis: Moving speed should be higher than walking speed, connection may be intermittent due to nodes missing. This might be caused by the shorter timespan where the smart phone is visible to the access point. The total travel time of the path should be much less than the time of a pedestrian.

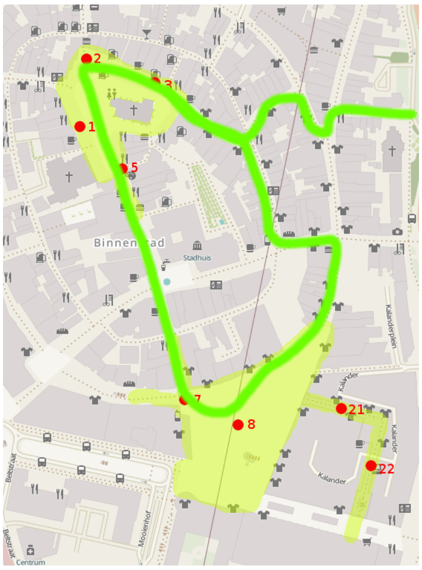


Figure 10: Route for the bicycle scenario

T3: Shopping
The shopping scenarios focus on short visits to the nearby shops on the Van Heekplein square. The visited shops are on the square itself or the Klanderij, which is next to the square. A typical market day will be included, to observe the increased population on the square when market stands are present.

Brick and mortar store
Objective: Enter shops near market place such as Jack & Jones and Primark. A possible recorded route could be *(8) -> — -> (8)*
Hypothesis: Phone disconnects as it enters a shop, and returns to the same hotspot after passing some time. The visitor remains on the same square.

Klanderij mall Objective: Visit the Klanderij mall, entering the H&M shop and Mediamarkt, for example *(8) -> (21) -> (22)* Hypothesis: Phone switches to hotspots in the Klanderij and leave the network whilst in shops. After returning to the Klanderij the session should be resumed at the previous connected node.

Market day
Objective: Visit market place, for example *(8) -> — -> (8) -> (8)*
Hypothesis: Phone remains on the market square, but might drop connections as it circles around the AP and switches radio. Phone should never connect to access points far from (8).

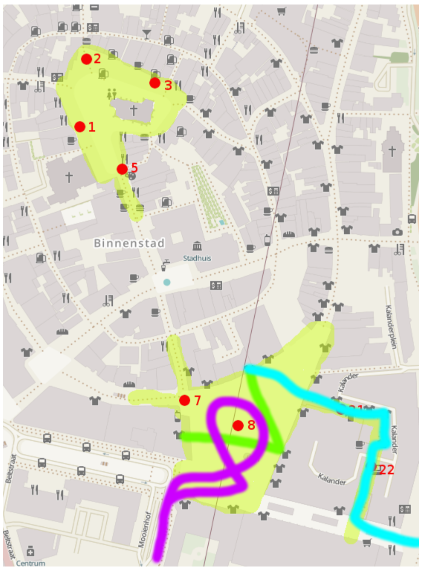


Figure 11: Route for the shopping scenario

T4: City entry points

City entry points are the main streets leading towards the squares. The routes are picked to cover different entries to a square with respect to the access points. In case a square has more entries with the same first node, such as north east to Van Heekplein, the route is omitted. The main goal is to see whether traces include all APs up to the square or start directly on the square itself. It is expected to see the bus stop, Klanderij and Kruitvat access points first. The city scenario is displayed in Figure 12.

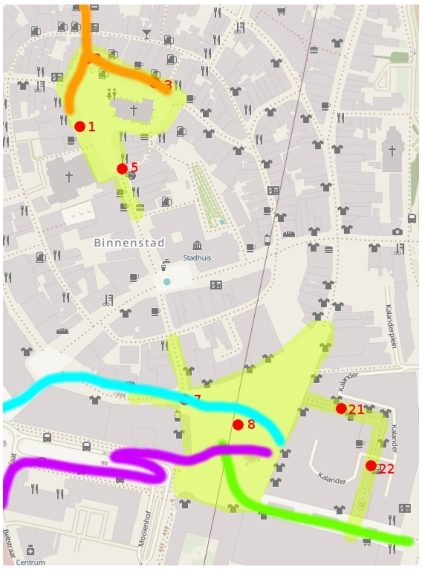


Figure 12: Routes for the city entry scenario

Entry south-east
Objective: Entry of city via south-east corner, for example from AH to market *(22) -> (8)*
Hypothesis: Phone connects quickly to 22 as its visible near the Albert Heijn. They alley has no other APs until user reaches market point (8).

Entry south-west
Objective: Entry of city via south-west shops, for example Primark to market square *(7) -> (8)* Hypothesis:
Phone connects to AP in or near Primark, and switches to market square.

Entry south-west bus
Objective: Entry of city via south-west bus stop, for example *(54) -> (8)*
Hypothesis: Phone connects to the local hotspot near the bus station as the bus arrives, and switches to the market AP upon entry.

Entry North
Objective: Entry of city via north point, for example *(2) -> (3) or (2) -> (1)*
Hypothesis: Phone connects to the local hotspot near the bus station as the bus arrives, and switches to the market AP upon entry.

5.2 Visitor counter

The visitor counter solely operates on the NDIX dataset. It creates paths using the aforementioned method and counts the number of unique MAC addresses in the city for a given day.

5.3 Event detection

The input for this experiment is the dataset of April 2016. The prototype analysis one specific month and outputs statistics on visitors and popular hotspots. By finding exceptional attendance in the city possible events are suggested.

5.4 Detecting shopping behaviour

The shopping behaviour uses the NDIX dataset, added with traces for reference validation. Shopping is classified by mobility and route through the city.

5.5 Mobility classification

The mobility classification experiment aims to find a baseline for walking and bicycle speeds in the city.

5.6 Raspberry PI Reference data

The Raspberry PI reference data is recorded in parallel with the traces for mobility. The added information is GPS positioning for reference tracking and high frequency recording of all visible access points.

6 Results

This chapter discusses the methods to process the experimental results, observations during the experiments and aims to answer the research questions using the results.

6.1 Experimental result notes

The additional experiments in the city produce traces for the prototype analyser. Some interesting remarks of the traces have been observed during the processing of the information. The network Enschede_Stad_van_Nu not present in any trace. Although the phone set for this network was connected it might have not established a connection due to the captive portal. It is interesting to note that the device was no longer present in the NDIX database. Leaving the phone in an idle state was part of the experiment, as visitors may not always unlock their phone to connect to the public WiFi infrastructure. The traces delivered by NDIX of this particular device did include previous traces of the exploratory experiment, meaning the MAC was correct.

The HTC Desire is not present in the database. This confirms devices which are disconnected are not tracked by the WiFi system. Therefore, WiFi sniffing is not one of the inputs to follow devices in the city centre.

The iPad has only a few traces over the days. It looks like the device only connects to the network when the display is actually on. It is unknown whether this behaviour is specifically for the iPad or includes other iOS devices.

The Nexus 6P and Raspberry Pi are best tracked devices, and produced the most traces. Both devices were in active use during the experiment for time keeping and reference tracking. Following these two the galaxy S4 is tracked in most places during the experiments.

Both days were tropical days in the Netherlands. With a temperature nearing the 30 degrees Celsius visitors may have opted to avoid lengthy walks in the city. The population on Tuesday was noticeable larger than on Wednesday.

The time between sessions - measured gap between leaving and joining - is measured negative on some occasions. It seems session remain active for much longer than the actual device is connected. In some cases the previous session ends more than ten minutes after the next session had begun. This behaviour of the network hugely impacted the mobility detector based on the gap between sessions. The detection is adjusted to use the session time, effectively using WiFi nodes as timestamp mark in the mobility path.

6.2 Go analyser

For prototyping and preliminary analyses the dataset is reduced in size by selecting individual months to run as partial set. April is the most dominant used month as it could be used for comparison between years. Other months are tested to see different patterns. The programme reads the dataset file and feeds all records to a processing pipeline.

The analyser pipeline consists of three stages, displayed in Figure 13. The pre-processing formats dataset rows and traces into generic path segments, connecting two consecutive sessions recorded. Path segment represent the transition between sessions. A segment preserves important characteristics of a transition. These characteristics are the source

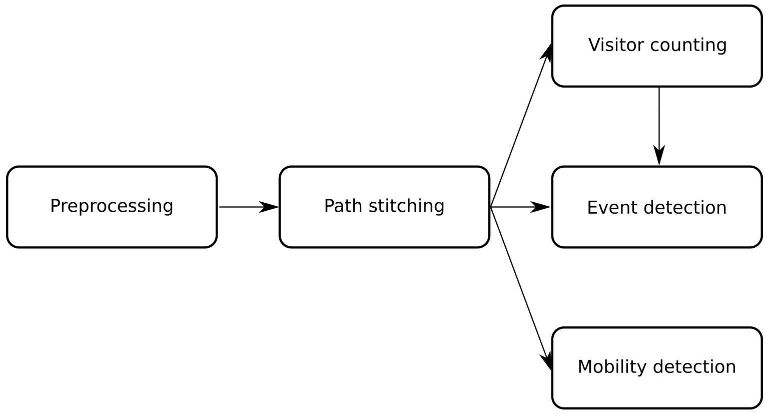


Figure 13: Three stages of WiFi data parsing

and destination access point name, segment start time and segment length. A parameter in the programme defines whether a segment starts from the leaving timestamp of the first access point or the start timestamp of the first access point. Using the leaving and entering time is preferred, as it measures the gap between sessions and travelling speed regardless of the activity during the session. If a visitor with bicycle travels fast from shop A to shop B, but spends an hour at the cafés the path should be marked at bicycle due to the fast movement and not pedestrian based on the long pause in one session.

Using these metrics the second stage stitches routes based on all sessions. The program has input parameters such as maximum time between sessions in order to be linked. This stage tries to find all paths in the set of sessions. All paths are stored in a hash map with MAC address as lookup key. For small timespans such a month of data the memory usage is acceptable and it gives an enormous speed-boost in further analysing of all paths.

Finally the third stage generates statistics based on the paths. These statistics could be number of visitors, popular areas of the city and detected events based on increased population.

The prototype tool is written from scratch in Go as a learning experiment with a new language. Go introduces channels embedded in the language to ease communication between components. Using channels a pipeline running in parallel processes is easily created. Channels transfer messages between components, accepting multiple input types such as the dataset and traces, and starting the event detector and mobility detector.

In this design with a pipeline the native channels support strong and clear communication between the stages and allow semi concurrency. All messages from stage 1 to 2 are sent directly between the components. Messages originating from stage 2 can be destined towards multiple statistics generators. To facilitate this a fan-out is inserted to duplicate the messages and spread them to all active generators. The generators process in parallel using go functions.

6.2.1 Programme parameters

In the programme multiple parameters for performance tuning and analysis configuration are available. In main.go static parameters are defined, these should remain the same in multiple runs and define general behaviour of the tool. In Listing 6 the parameters are listed.

Listing 6: Excerpt of main.go

```
maxRows                = 200000  // Maximum number of rows to
// parse, fail safe for testing.
// Set to high number to disable.
maxTimeBetweenSegments = 15 * 60 // Link all segments with less
// than 15 minutes interval

pathDirectional = false // Store paths directional or clustered
// segments regardless of traveling direction
pathEndToStart  = false // Measure from last end to next start
// time between sessions

// Defines the hour at which a section of the day begins
timeBeginMorning   = 6  // morning begins at 06:00
timeBeginAfternoon = 12 // afternoon begins at 12:00
timeBeginEvening   = 18 // evening begins at 18:00
timeBeginNight     = 0  // night begins at 00:00
```

In classifier.go the parameters are more dynamic. These are trained by the model and may vary per run. The weights change the influence of the feature on the score of the segment. The following settings are defaults. These are picked to give much weight to velocity and set the mobility status to bicycle at 2 metres per second, about 7.2 kilometres per hour.

The minimum session time is included to reduce the effect of quick hotspot swapping and creating an artificial super speed when hopping between nodes. In experiments a swap of less than 2 seconds is found. Given the distance between hotspots is at least a hundred meters, this gives a velocity of 50 metres per second, not a realistic value for pedestrians. The parameters are listed in Listing 7.

Listing 7: Excerpt of classifier.go

```
weightTrainedSkip        = 1.5 // Weight for the not is skipped feature
weightTrainedVelocity    = 1.0 // Weight for the velocity computed
scoreThresholdBicycling = 2.0 // Threshold for classifying bicycling mobility
minimumSessionTime       = 20  // Minumum session time to compute velocity
```

6.3 Reference data

The dataset provided by NDIX includes many sessions between 2015 and 2016. Prior to use this dataset as reference input some experiments were conducted to visualise the contents of the set. Finally, the visualization of the city counts is in an interactive HTML map. This map show links between all access points and pop-up with the total number of visitors for that particular link.

6.3.1 Average 24H split

Diving a bit deeper into the dataset using Pandas, a python library for data analysis, a clustering is created per access point and day listing the number of sessions. Figure 6 contains a barchart showing all access points and the number of sessions in the dataset.

The most connected access point is located on the largest square, the Van Heekplein. Access point #8 has close to double the amount of connections to the second largest node.

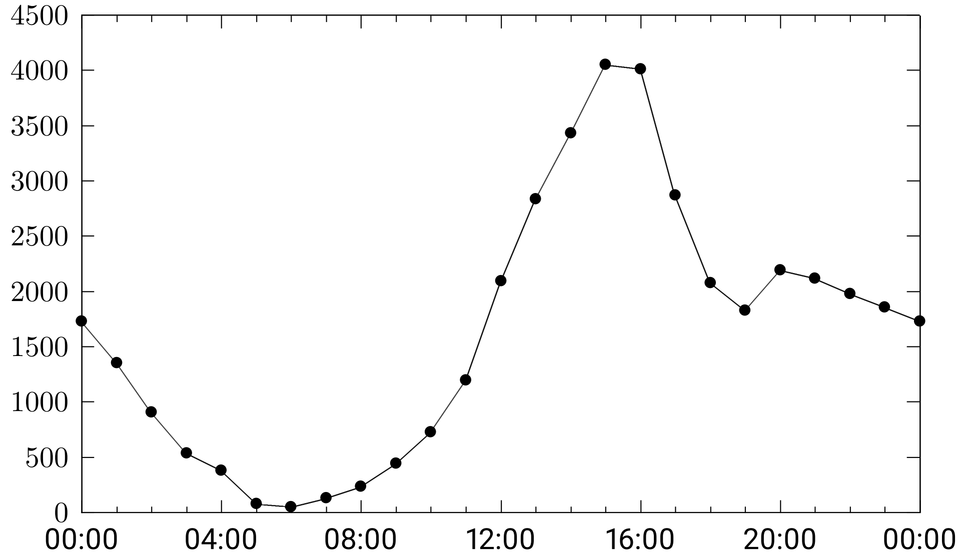


Figure 14: Visitor count for a typical day averaged over April

Looking at the session times it is easily visible what the peak hours of the city are. Figure 14 visualises the peak hours of a typical day in April in Enschede. A typical day means all weekdays and weekend-days accumulated over April divided by the number of days in April. Most visits are in the afternoon up to early evening. A small increase after dinner time may indicate people eat at home and join later in the evening to visit pubs in the town.

6.3.2 Preliminary experiment

Acquiring the correct set for the experiments is vital to the results of this work. Preliminary discovery of the WiFi network revealed that only devices actively connected are logged and sessions are stored as a summary of the total time connected per access point. Roaming between access points is stored as separate sessions.

During this thesis work NDIX provides WiFi traces as supplement to the dataset. A new dataset needs to be created at an external party to anonymise the records. Although NDIX would like to create a net set, due to external dependencies it is unlikely to extract a full new dataset of 2017. Instead NDIX offers WiFi traces of individual devices. To find differences between traces and the full set a small experiment involving multiple phones in the city moving about is set-up.

At the time of this experiment accesspoint ENS-AP-06 is removed for construction work in the building it was connected to. This means between ENS-AP-05 and ENS-AP-07 the network is no has full coverage. This experiment revealed that smartphones not connected to Eduroam or Enschede_Stad_Van_Nu do not produce trace results. This gives doubt to the statement that the system logs all visible devices, even when not connected to the local WiFi network. Furthermore the traces are less detailed, as the contain less information per record. The information stored in the records still includes the node it connects to and session length, information used by the system. In Table 6 the full trace record is explained.

Table 6: Metrics recorded in each trace using the NDIX WiFi network

Column	Metric	Description
1	Access Point MAC Address	MAC of the connected device
2	Access Point Hostname	The internal name of the connected access point
3	SSID	The name of the connected WiFi network
4	Association Time	Start time in unix timestamp format
5	Disassociation Time	End time of the session in unix timestamp format

6.3.3 Visualizing city dynamics

The visualization of the city dynamic uses Leaflet in an HTML page. Modern browser libraries have extensive support for geospatial display options. In this viewer all access points in the dataset are plotted and connected by a network of routes walked. A map of April 2016 is displayed in Figure 15. Visitor counts are displayed on this map with links representing the number of devices passed on that segment. By selecting one of the access points or links a pop-up show all traffic passing on on that particular city area.

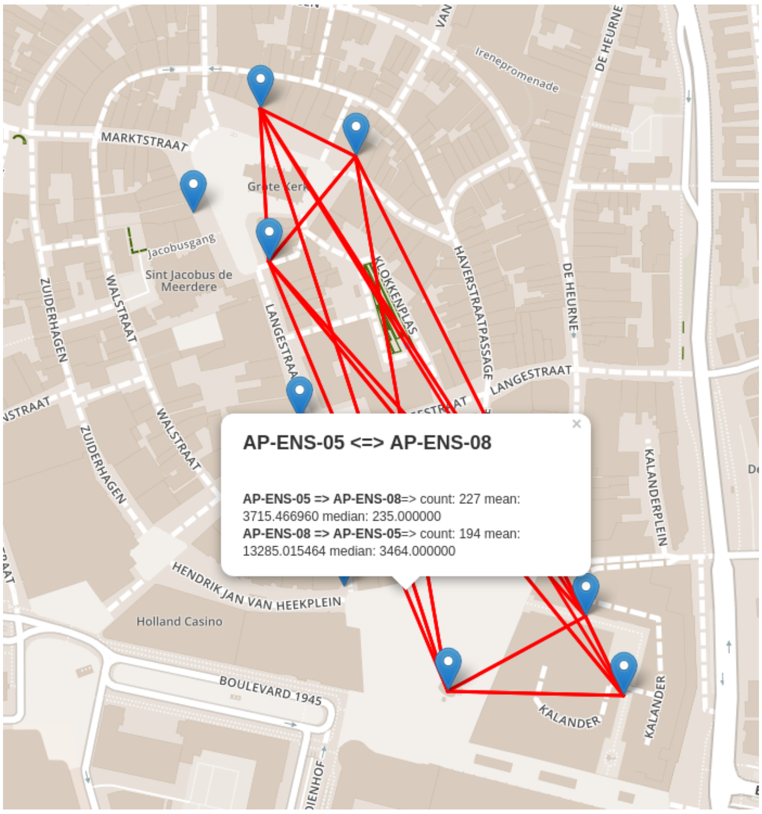


Figure 15: Overview city dynamics in April 2016. Plotted lines represent the average afternoon of April.

The current implementation reads the exported statistics of the Go tool. A database storing the intermediate results would be necessary to visualize different timeframes than the current export of the tool.

6.4 Visitor counter

The visitor counter iterates over all path objects and increases daily counters for predefined areas - clustering nodes - and time segments such as morning, afternoon, evening and night. Each day is split in these four segments to differentiate daytime shopping and relaxation in the evening. This split is visible in Figure 16.

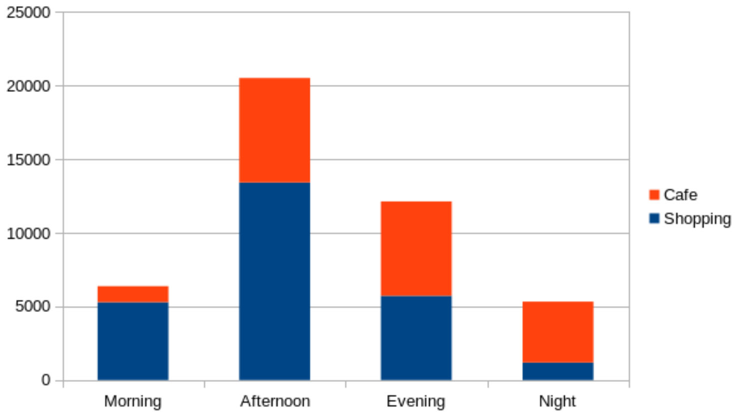


Figure 16: Split of activity type per day time. Cafe is the northern square with pubs and cafés. Shopping is the south square with shops.

The resulting count is a rough representation of the actual population in the city centre as only connected devices are counted. The counting enables overview maps such as in Figure 15, event detection based on anomalies and long term population trends. A long term view of all counted visitors is in 17.

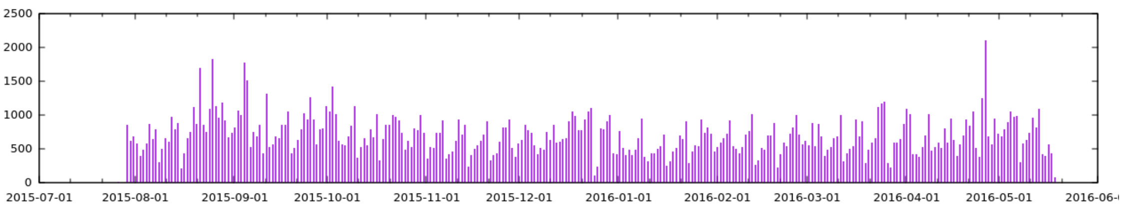


Figure 17: Overview of visitor count throughout the year, showing repeating patterns representing weekdays and weekends.

This graph shows a trend with weekly repeating patterns for busy Saturdays and closed shops on Sunday. Yearly trends are visible as well, the summer months display more visitors than the winter months. During Christmas and new year the number of visitors drop to almost zero.

6.5 Event detection

Using the generated list of paths representing visits in the city the event detector scans for anomalies in the visitor counts. Anomalies are days with a much lower or higher attendence detected by the system. For this experiment the threshold difference is set to $sigma = 2$. Any day with less or more than the moving average + standard deviation is marked as anomaly.

The following table lists all events detected using the full year as input of the detector.

Table 7: Overview of detected events in Enschede

Date	Count	Event
2015-07-30	848	First day in dataset
2015-08-22	1696	UT Kick-In
2015-08-26	1820	UT Kick-In
2015-09-05	1765	—
2015-09-06	1502	—
2015-09-12	1313	—
2015-10-03	1418	Celebration of reunition of East and Western Germany
2015-12-25	103	Christmas
2015-12-26	226	Christmas
2016-04-25	371	—
2016-04-27	2107	Kingsday

The days listed in Table 7 are larger events in Enschede. One of the noteworthy events is the celebration of the reunition of East and Western Germany, many Germans opt to visit Enschede on this day as all shops in Germany close. The binnenstadsmonitor enschede noticed this event as well with over 80% of the cars in the parking garages were German (Scholten, van de Wiel, & Seker, 2017).

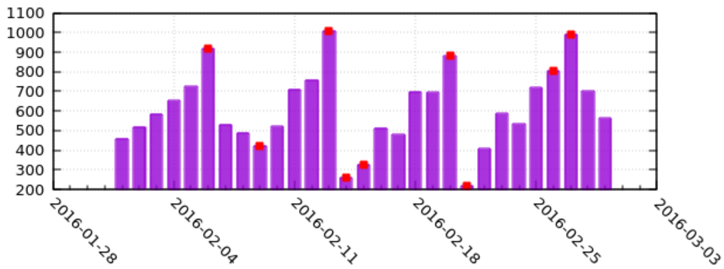


Figure 18: Overview of visitors in februari for local temporal detection. Red dots indicate anomalies detected by the algorithm.

Looking as small temporal scale the busy days of the week are revealed. For example looking at Februari 2016, the counting mechanism shows that Saturdays are in general more crowded in the city whereas Sundays are very low attendance. This weekly pattern of busy Saturdays and quiet Sundays is described in the Enschede stadsmonitor 2016 (Scholten et al., 2017). Februari is shown in Figure 18, where red dots represent detected anomalies. All shops are closed on Sundays, so it is expected to see less attendance in the city. Saturday is the market day, and attracts larger groups. In Table 8 detected events for Februari are listed.

On small scale it is visible that weekdays are the average and weekends stand out in terms of attendance. One interesting observation is the listing of 2016-02-09 as a low. This is a Tuesday and is a market day, in general should bring more visitors to the city.

6.6 Detecting shopping behaviour

Shopping behaviour detection depends on the clustering of paths. By concatenating segments a full path of a user is established. The input for the shopping detection is the dataset filtered for access points located near the Van Heekplein. The included access

Table 8: Overview busy days in Februari 2016 in Enschede

Date	Day of week	Count	Detection
2016-02-09	Tuesday	422	low
2016-02-13	Saturday	1010	high
2016-02-14	Sunday	261	low
2016-02-15	Monday	328	low
2016-02-20	Saturday	880	high
2016-02-21	Sunday	220	low
2016-02-26	Friday	807	high
2016-02-27	Saturday	990	high

points are "AP-ENS-07", "AP-ENS-08", "AP-ENS-21" and "AP-ENS-22". Access points "AP-ENS-53", "AP-ENS-54" are not included as they are not recorded in the dataset.

The distribution of path lengths in seconds as duration is listed in Figure 19. In total over 46 thousand paths are included in the shopping detection. The distribution shows that most paths created are around two minutes in timespan. This means that visitors either travel directly to their destination to visit or the path metadata is inconclusive for tracking different types of shopping.

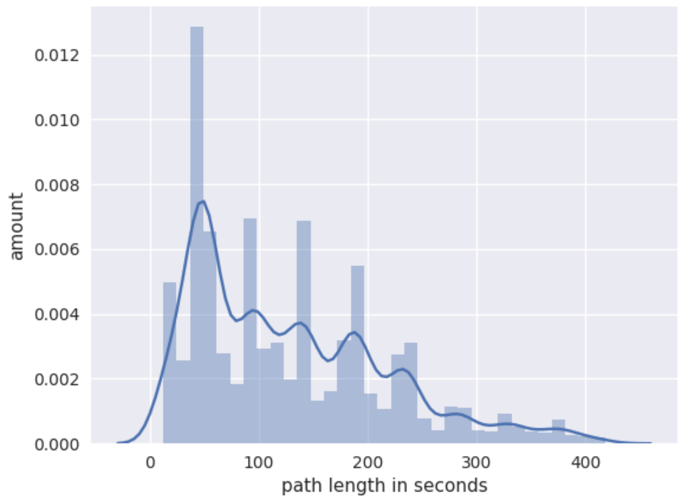


Figure 19: Distribution of path lengths in shopping behaviour detection. $n = 10582, min = 12sec, max = 418sec, avg = 128.2sec, std = 88.7sec$

To improve this detection of shopping the system is extended to compare paths recorded on the same day. This gives a roughly 50/50 split in visitors whom connect multiple times on one day. The system found 5388 visitors who reconnected to the network after the initial path was stopped. A total of 5778 visitors did not reconnect on the same day after the first path stopped. This means that roughly 51.7% of the visitors only connects for an average of 2 minutes per day in the shopping area.

The total connected time during a visit in the city is roughly four minutes. A total shopping time of four minutes does indicate the average visitor quickly retrieve an item in the city, or spends a short amount of time on the main squares connected to the WiFi network.

As four minutes of shopping time is very short, another method is proposed by taking the first and last connection of a day. This gives the following distribution as displayed in

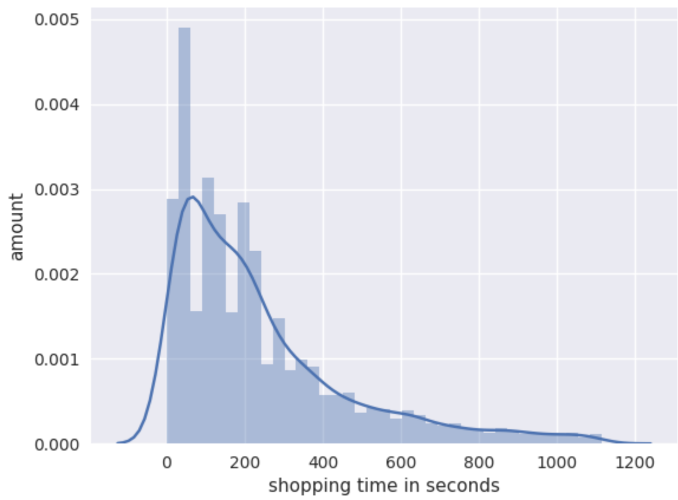


Figure 20: Distribution of single connection length in shopping behaviour detection. $n = 4683, min = 1$sec$, max = 1114$sec$, avg = 240.4$sec$, std = 227.4$sec

Figure 21. On average visitors spend up to one hour between their first and last connection in the city. One cause for this might be to only start WiFi after some shopping has been done to find other activities in the city.

The total time per day is much longer than the connected time per day, suggesting that visitors lose their WiFi connection during shopping. This could be caused by the indoor shops which have no coverage of WiFi. If this happens, it could mean visitors rather enter shops than walk outside for window shopping.

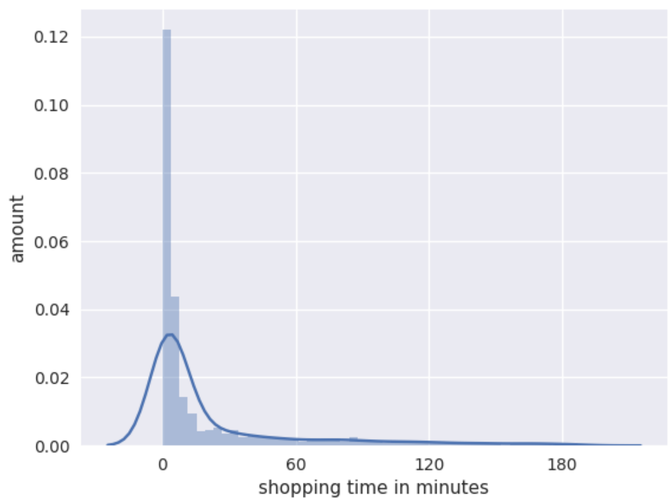


Figure 21: Distribution total time on one day. Takes the first connection time and last connection time of the user per day. Time is in minutes. $n = 2171, min = 0.1$ min$, max = 190.7$ min$, avg = 21.9$ min$, std = 38.3$ min

6.7 Mobility performance

The performance of the mobility detection is hugely dependent on the accuracy of the network session data. As the sessions are recorded much longer than the actual active time, the gaps between sessions are not usable for mobility detection. The negative times and variation simply creates to much uncertainty to conclude anything based on gap alone. By using the start time of both session, including the session time of the first session, the performance is increased.

A reason to use session start times only and not the gap between sessions (thus leaving and joining the next) is due to that sessions on different nodes may overlap. In a few experiments the previous session lasted up to ten minutes after joining the next node. This means the session end time stamp may be later than the session start of the next one. The exact motivation for this behaviour is unknown, but due to caching on the access points session may be kept open for a short while after losing connection to a device. The access point nodes are not communicating with each other, so there is no management layer moving sessions between access points.

However, as the session time is included this influences cycling detection. For example, a fast moving bike may pause and connect to the network at a certain point. As it connects this triggers a long duration in one of the sessions in the path. As the session is lengthy and path are often only a few segments long, the mobility detection detect pedestrian movement instead. An example of a trace with varying scores is listed in Listing 8.

Listing 8: Example cycling trace with varying scores

```
Score              Gap Starttime                     Path nodes
0.7019204064608325 172 Wed Jul 19 15:50:26 CEST 2017 {AP-ENS-54 -> AP-ENS-07}
2.4799004123801556  40 Wed Jul 19 15:53:18 CEST 2017 {AP-ENS-07 -> AP-ENS-08}
1.7428353559315313 237 Wed Jul 19 15:53:58 CEST 2017 {AP-ENS-08 -> AP-ENS-02}
1.1324649902240178  75 Wed Jul 19 15:57:55 CEST 2017 {AP-ENS-02 -> AP-ENS-01}
5.879136790703723   62 Wed Jul 19 15:59:10 CEST 2017 {AP-ENS-01 -> AP-ENS-07}
2.4799004123801556  40 Wed Jul 19 16:00:12 CEST 2017 {AP-ENS-07 -> AP-ENS-08}
6.322962712774866   24 Wed Jul 19 16:00:52 CEST 2017 {AP-ENS-08 -> AP-ENS-53}
Avg score  2.962731582979326
```

The performance for all tested scenarios is displayed in Table 9. Note that the number of testcases is lower than the scenarios listed in Section 5. The parameter to link all sessions with less than 15 minutes time difference creates longer paths than the experiments describe. One of the walking paths is 16 sessions in length, and traverses all nodes of the network.

Table 9: Performance of mobility detection. Displaying correct detected, false positives, false negatives and per cent of correct detected.

Scenario	Number of testcases	Correct detected	FP	FN	%
Pedestrian	8	3	1	4	37.5 %
Cycling	4	2	2	0	50 %

6.7.1 Raspberry PI Reference data

The dataset provided has limited input for the mobility detector. Detailed movement in the city is not captured using the network. By enriching the dataset using recordings of the Raspberry Pi, the movement detector is reviewed a second time. This shows the difference in detection options.

The data recorded by the Raspberry Pi stores a record each second, additionally to the NDIX dataset it stores GPS positions. Unfortunately direct mapping of network MAC to recorded MAC is not available. The recorded access point addresses are different than the addresses seen in NDIX dataset. The vendor part of the addresses match, however, the full MAC address is different. This may be caused by the different radios in each point broadcasting different addresses.

To circumvent this problem an ideal scenario for WiFi tracking is created. Using the GPS positioning of the recorded data and geofencing with known location of all access points for each record in the dataset the location is mapped to the nearest access point. In this ideal world two different approaches for following a access point are used. First, the strongest recorded point is used to track RSSI of the AP. Using the GPS and AP location the distance is calculated. This distance is used as a measurement for input in the mobility detector. In the second approach the connected access point is used. In some scenarios where multiple access points could have been connected, it may not be the nearest one. In this scenario the RSSI is again linked to the GPS distances and reused in the mobility detector.

Having one record per second gives a predetermined interval for data processing. The distance between records (based on GPS) is used as a indication of velocity. This helps the mobility detector finding the local motion in a trace, and gives a more fine grained look in the local mobility between access points. This is essential to detect whether visitors move from shop A to shop B, or walk from shop A to shop B whilst stopping and looking through windows of other shops. Thus, enabling the system to differentiate targeted shopping and window shopping.

The performance of the reference data is better than the performance on traces. Note that the number of test cases is larger, this is due to more datapoints per trace and disconnection of traces when GPS signal is lost indoor. The pedestrian cases especially improve by the emulated ideal world due to much richer traces. However, in the results this is not reflected as indoor situations are worse. In an ideal world indoor should perform as good as outdoors, but is not realised due to the use of GPS as reference.

Table 10: Performance of mobility detection using RPI. Displaying correct detected, false positives, false negatives and per cent of correct detected.

Scenario	Number of testcases	Correct detected	FP	FN	%
Pedestrian	20	15	1	3	75%
Cycling	4	3	0	1	75%

7 Discussion

This work explores the WiFi network infrastructure in Enschede for applications in smart city crowd management. By counting visitors and visualizing mobility patterns new insight in crowd movement is created. However, the visitor counting is only a device counter. This is useful when comparing relative visitor numbers between days. It may not reflect the actual number of people in the city due to many devices not connected to the network or simply visitors without any devices at all. This work has no attempt of comparing actual visitor counts in the city and computed number of visitors by the prototype programme. Due to lack of reference material and focus on dataset analytic this step is missing.

7.1 Research questions

Is the data sufficiently accurate for counting visitors in specific locations of the city?

The data is accurate enough for counting the relative number of visitors in a specific location such as one of the squares. The number of devices present indicate the population present. Each access point covers a specific area of a maximum of a few hundred meters. By grouping the nodes the count of a region, such as a square, is possible. However, the relation between number of devices connected to the network and actual visitor count is not established. The data shows increase of visitors during events, and thus enables crowd management by comparing day to day fluctuations. Exact counts of visitors is not yet investigated.

How visitors move within the city centre? Where are the popular places? How people commute within the city centre?

The mobility within the city centre is difficult to extract from the dataset as explained in the experiment with mobility. The mobility pattern detector works best when evaluating the gaps between sessions, eliminating short stops between visits. As the mobility detector now uses the start time of all sessions at nodes, it may include shopping time and falsely label bicycles as pedestrians. For example, if someone visits a shop and remains there for a couple of minutes, this impacts the travel time. A longer time indicates slow movement and is classified as pedestrian. The 66% score on the cyclist detection might be hugely reduced using a different dataset if motion is not continual between points.

Popular places in the city shift during the day and night time. In the morning Van Heekplein is the most popular area to be, near all shops and possibly market on a Tuesday or Saturday. Nearby access points show the same pattern with regular shoppers connecting to the network. The Old Market is more popular at evenings and night. At this time all the cafés open and people visit them for a drink and dinner. In the morning at the Old Market during April the system counted an average of 1149 devices. The median session time is 235 seconds. This indicates many counted devices are passing by the system and not staying at the cafés. During the evening the number of devices is 1381, with an median session length of 2041 seconds.

What transportation methods do visitors use when entering the city?

The network set-up at the time of the dataset creating makes it hard to distinct visitors based on entry points. In experiments it is found that southern entrance is divided by entry via the bus stop, Klanderij and Primark. However, due to the missing access points on the bus stop, the entry by public transport, bicycles and pedestrians are equal.

What privacy extensions to the dataset could help minimise personal identification?

In this work the MAC addresses are not reused to track a specific person. The addresses are used to create paths based on recorded segments. A requirement to create these paths is that the addresses remain unique and assigned to a specific visitor during their stay in the city. The recycling of addresses could happen more frequently. The dataset has no recycling of addresses and the to be generated sets will recycle every month. This could happen daily given that most visits in the centre of the city are not overlapping multiple days. Tourists and other visitors will have to sleep outside of the WiFi coverage in the city centre. If the cycling of MACs is more frequent, the system will no longer be able to track recurring visits. This is something to keep in mind when deciding the balance of privacy of users and crowd management in the city.

What useful information the WiFi sniffers provide for analysing and observing visitors behaviour and mobility patterns in the city centre? The WiFi sniffers in the city have proven follow connected devices only. Sniffing as in following all devices passing by the access points is not recorded in any of the data sources. Give that all devices must be connected, it reduces the number of visible devices greatly.

The useful information is the sessions to indicate the number of visitors in the city and crowded areas as all nodes are linked to a specific area of the city centre. It gives insight in shopping times and café & bar hours. The mobility using this network is difficult, sessions are not coherent and do not indicate when a device left a specific area of the city. Having more accurate sessions should help measuring the gap between hotspots and the actual moving velocity of devices, thus mobility in the city.

The retrieved paths and detected events are in agreement with the observations of the local council presented in the binnenstadsmonitor 2016. This summary of visitors displays the same weekly trend with building attendence to the weekend, a peak on Saturday and quiet Sunday as shops close. The relative counts between the northern cafe square and southern shopping square are in agreement as well. The binnenstadsmonitor shows that much more visitors enter the large square for shopping and move about that section of the city (Scholten et al., 2017).

7.2 Dataset notes

This section discusses some notes on the dataset and experiments conducted for the research questions.

7.2.1 Session inaccuracy

Although the implementation in Go gives a speedy counting tool and rough estimations of visitors, the prototype lacks refinement in mobility detecting. The current network coverage only includes one of the many routes between squares. As nodes may get skipped when traversing squares, it is not known whether someone was moving slowly on the tracked road or travelling fast on a side road taking a longer route. For more accurate modelling more recorded points are necessary, and require expansion of the WiFi coverage. The lack of accurate session information hugely impacts the accuracy of mobility detection. Fast moving devices with pauses are detected as slow moving pedestrians whilst the actual moving speed may have been high.

7.2.2 Data detect granularity

The WiFi network also imposes a limitation on accuracy of crowd following. Recorded sessions do not include the RSSI over time. This is needed to create a more detailed model of motion and overview of micro mobility on squares. This prototype only knows if someone is present on a square and the duration of the stay. Exploiting more detailed information of sessions could open options to detect groups and possibly nuisance in the city. It could establish paths on the squares, often visited market vendors and micro dynamics within a hotspot zone.

7.2.3 Experimental sample size

The experiments and training set for the mobility detection include a small sample set. To completely validate the prototype more input should be tested. The experiments were designed having in mind a new dataset of 2017 would be generated and available for testing. Especially a set covering the traces should be interesting to compare. The labelled traces should be included in the set, and similar records will be annotated with the same labels by the prototype.

7.3 Python versus Golang performance

The start of this work was developed using Python as a familiar language for data science and large set processing. the initial dataset discovery uses Python and its rich ecosystem of libraries to visualize important aspects of the dataset, such as distribution over access points, time length of sessions and global spread of sessions over time.

After the initial discovery it was opted to use Golang for further development due to its high performance nature and long loading times in Python. The NDIX dataset takes over 10 seconds to load in Python, whereas Go can load the set in less than 200ms. This turnover rate of Go helps with the development of the algorithms and testing as less time is spend waiting on results of the algorithm.

However, as the project progressed it became clear that Go had its own disadvantages. For example, the libraries provided by Go are not as rich as Python offers. Visualisation can be performed using other tools such as gnuplot, thus minimizing dependencies on graphical libraries. Although gnuplot offers good graphs, the processing of results and visualizing distributions is not trivial. For this reason the project reverted back to Python and accepted the long runtimes. At this point Go was able to load and analyse the dataset including counting and detection in 300 ms. The same programme in Python took over 30 seconds to complete.

8 Conclusion

Smart cities are a interesting area of research. In this first exploratory step the WiFi network in Enschede is exploited for smart city applications. By observing visitors and their mobility patterns new input for crowd management is created. This increases options for safety by taking preliminary actions if squares get to crowded. Furthermore the system reveals interesting information for advertisement as it shows the effects of running advertising campaigns by showing the increase of visitors on a specific day compared to a normal day.

The current implementation of mobility detection needs refinement and should benefit from more accurate network information. Some sessions are recorded up to ten times longer than the actual duration of the connection was. Using the additional RPI data for classification more detailed classification was possible, and the detection rate was improved. This local motion is necessary to increase the detail of tracking in the large data set. Something between the session of minutes and tracking one datapoint per second should proof te be a good starting point for accurately determining the motion of visitors.

The prototype programme proofs to be a working concept with compartmented blocks. By reusing previous computed intermediate results the statistic generators transform sections and paths into visitor counting, event detection and mobility detection. These three metrics should help in urban planning as they reveal the dynamics of the city.

8.1 Future work

Future work depends on the extensions of the WiFi network. The current prototype has limited transport method detection as the system relies on speed and path of visitors. A fast traversed route is noted as a bicycle and slower travelling people as pedestrians. The last group is combined of visitors at their destination (window shopping, actually visiting the shop or sitting at a café) and slow moving people. Having more detailed records or more frequent sample points - effectively more access points - the route segments are smaller and more accurate with respect to the actual movement. By extending the network with more access points more streets could be covered and alternative routes linking the squares could be monitored as well.

The current implementation utilizes a few infrastructure specific information as possible. It is able to scale with more access points and only requires the location to determine distance between nodes. The actual placement and inclusion in the system does minimally affect the visitor counting or event detection. The clustering of nodes for regions should be updated as well in order to add new access points to current regions or define more regions.

By including the bus stop and train station the pedestrian group could be distinct in public transport users and automobile / living nearby visitors. The detection of automobile entries in the city is difficult as the users enter the WiFi system like other visitors, there is no separate access point or information for this class of visitors. If sessions include more detailed signal strength information, visitors of the parking garage in the Klanderij could be separated based on the stronger signal as they emerge from the underground parking.

References

Blanke, U., Tröster, G., Franke, T., & Lukowicz, P. (2014, April). Capturing crowd dynamics at large scale events using participatory gps-localization. In *2014 ieee ninth international conference on intelligent sensors, sensor networks and information processing (issnip)* (p. 1-7). doi: 10.1109/ISSNIP.2014.6827652

Calzada, I., & Cobo, C. (2015). Unplugging: Deconstructing the smart city. *Journal of Urban Technology*, *22*(1), 23-43. Retrieved from `http://dx.doi.org/10.1080/10630732.2014.971535` doi: 10.1080/10630732.2014.971535

Caragliu, A., Bo, C. D., & Nijkamp, P. (2011). Smart cities in europe. *Journal of Urban Technology*, *18*(2), 65-82. Retrieved from `http://dx.doi.org/10.1080/10630732.2011.601117` doi: 10.1080/10630732.2011.601117

Faragher, R., & Harle, R. (2014, 01). An analysis of the accuracy of bluetooth low energy for indoor positioning applications. , *1*, 201-210.

Gong, S., Loy, C. C., & Xiang, T. (2011). Security and surveillance. In T. B. Moeslund, A. Hilton, V. Krüger, & L. Sigal (Eds.), *Visual analysis of humans: Looking at people* (pp. 455–472). London: Springer London. Retrieved from `https://doi.org/10.1007/978-0-85729-997-0_23` doi: 10.1007/978-0-85729-997-0_23

Helbing, D., Johansson, A., & Zein Al-Abideen, H. (2007, January). The Dynamics of Crowd Disasters: An Empirical Study. *ArXiv Physics e-prints*.

Hollands, R. G. (2008). Will the real smart city please stand up? *City*, *12*(3), 303-320. Retrieved from `http://dx.doi.org/10.1080/13604810802479126` doi: 10.1080/13604810802479126

Mallah, J. E., Carrino, F., Khaled, O. A., & Mugellini, E. (2015). Crowd monitoring. In N. Streitz & P. Markopoulos (Eds.), *Distributed, ambient, and pervasive interactions: Third international conference, dapi 2015, held as part of hci international 2015, los angeles, ca, usa, august 2-7, 2015, proceedings* (pp. 496–505). Cham: Springer International Publishing. Retrieved from `https://doi.org/10.1007/978-3-319-20804-6_45` doi: 10.1007/978-3-319-20804-6_45

Pan, G., Qi, G., Zhang, W., Li, S., Wu, Z., & Yang, L. T. (2013, June). Trace analysis and mining for smart cities: issues, methods, and applications. *IEEE Communications Magazine*, *51*(6), 120-126. doi: 10.1109/MCOM.2013.6525604

Puel, G., & Fernandez, V. (2012). Socio-technical systems, public space and urban fragmentation: The case of 'cybercafés' in china. *Urban Studies*, *49*(6), 1297-1313. Retrieved from `https://doi.org/10.1177/0042098011410333` doi: 10.1177/0042098011410333

Sapiezynski, P., Stopczynski, A., Gatej, R., & Lehmann, S. (2015, 07). Tracking human mobility using wifi signals. *PLOS ONE*, *10*(7), 1-11. Retrieved from `https://doi.org/10.1371/journal.pone.0130824` doi: 10.1371/journal.pone.0130824

Scholten, A., van de Wiel, E., & Seker, H. (2017, 01). *Binnenstadsmonitor enschede 2016* (Tech. Rep.). Kennispunt Twente.

Su, K., Li, J., & Fu, H. (2011, Sept). Smart city and the applications. In *2011 international conference on electronics, communications and control (icecc)* (p. 1028-1031). doi: 10.1109/ICECC.2011.6066743

Verbree, E., Zlatanova, S., van Winden, K. B. A., van der Laan, E. B., Makri, A., Taizhou, L., & Haojun, A. (2013). To localise or to be localised with wifi in the hubei museum? *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XL-4/W4*, 31–35. Retrieved from `https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-4-W4/31/2013/` doi: 10.5194/isprsarchives-XL-4-W4-31-2013

A List of WiFi networks on Van Heekplein

```
date,latitude,longitude,MAC,SSID,QualityLevel,SignalLevel,NoiseLevel
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BD:53:95,eduroam,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BD:53:C2,Enschede_Stad_van_Nu,65,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BD:53:92,Enschede_Stad_van_Nu,61,0,0
2017/07/19 15:36:36,52.217893,6.897492,72:08:8B:16:08:CC,HG655D-1608CD,51,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BD:53:F5,eduroam,63,0,0
2017/07/19 15:36:36,52.217893,6.897492,74:B5:7E:26:8C:05,Lipstick,65,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BD:53:C5,eduroam,63,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BD:53:C3,\x00\x00\x00\x00\x00\x00,59,0,0
2017/07/19 15:36:36,52.217893,6.897492,6A:B5:7E:26:8C:06,KPN Fon,65,0,0
2017/07/19 15:36:36,52.217893,6.897492,50:60:28:61:97:80,Enschede_Stad_van_Nu,41,0,0
2017/07/19 15:36:36,52.217893,6.897492,50:60:28:61:97:82,\x00\x00\x00\x00\x00\x00,41,0,0
2017/07/19 15:36:36,52.217893,6.897492,FC:0A:81:2F:B9:22,etamgroep,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,50:60:28:61:97:84,eduroam,41,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BD:53:F2,Enschede_Stad_van_Nu,63,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:02:6F:F8:6C:E4,10-001,57,0,0
2017/07/19 15:36:36,52.217893,6.897492,50:60:28:61:97:83,\x00\x00\x00\x00\x00\x00\x00,43,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BD:53:F4,\x00\x00\x00\x00\x00\x00\x00,65,0,0
2017/07/19 15:36:36,52.217893,6.897492,FC:0A:81:2F:B9:23,etamgroep-i,41,0,0
2017/07/19 15:36:36,52.217893,6.897492,FC:0A:81:2F:B9:20,NMN-beheer,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BD:53:F3,\x00\x00\x00\x00\x00\x00,65,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BC:9B:F3,\x00\x00\x00\x00\x00\x00,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,FC:0A:81:2F:B9:24,CTX,39,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:02:6F:F8:6C:E5,10-002,55,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:02:6F:F8:6C:E6,10-003,57,0,0
2017/07/19 15:36:36,52.217893,6.897492,F0:5C:19:66:B0:60,JdGW1ReLeSs,51,0,0
2017/07/19 15:36:36,52.217893,6.897492,F0:5C:19:66:AF:E0,JdGW1ReLeSs,51,0,0
2017/07/19 15:36:36,52.217893,6.897492,56:BE:F7:19:91:86,Ziggo,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BD:53:93,\x00\x00\x00\x00\x00\x00,67,0,0
2017/07/19 15:36:36,52.217893,6.897492,5C:DC:96:35:DB:48,VGV751935DB48,37,0,0
2017/07/19 15:36:36,52.217893,6.897492,6C:AA:B3:24:FA:F8,Ziggo,59,0,0
2017/07/19 15:36:36,52.217893,6.897492,F0:92:1C:D6:89:29,HP-Print-29-ENVY 4500 series,49,0,0
2017/07/19 15:36:36,52.217893,6.897492,7E:E9:D3:40:3B:BC,Ziggo,51,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BD:53:C4,\x00\x00\x00\x00\x00\x00\x00,61,0,0
2017/07/19 15:36:36,52.217893,6.897492,84:D4:7E:CA:78:C0,AT-Devices,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,2C:5D:93:0D:70:98,Ziggo,61,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BC:9B:F2,Enschede_Stad_van_Nu,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,2C:5D:93:23:6D:38,Ziggo,53,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BC:9B:94,\x00\x00\x00\x00\x00\x00\x00,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BC:9B:F4,\x00\x00\x00\x00\x00\x00\x00,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BC:9B:F5,eduroam,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,88:03:55:8E:D9:CB,VGV75198ED9CB,51,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BC:9B:92,Enschede_Stad_van_Nu,49,0,0
2017/07/19 15:36:36,52.217893,6.897492,D2:B2:C4:54:89:8E,Ziggo,39,0,0
2017/07/19 15:36:36,52.217893,6.897492,84:D4:7E:CA:78:C1,41,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BD:53:94,\x00\x00\x00\x00\x00\x00\x00,67,0,0
2017/07/19 15:36:36,52.217893,6.897492,D0:B2:C4:54:89:8D,VivaLaDiva2017-normal,39,0,0
2017/07/19 15:36:36,52.217893,6.897492,84:D4:7E:12:5C:E2,ESPRIT_free_WiFi,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,B0:C2:87:77:9C:3A,Ziggo38131,47,0,0
```

```
2017/07/19 15:36:36,52.217893,6.897492,8A:15:04:8D:EA:9F,\x00\x00\x00\x00,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,FC:0A:81:08:DF:D0,\x00,53,0,0
2017/07/19 15:36:36,52.217893,6.897492,D4:76:EA:0B:0B:3D,H369A0B0B3D,55,0,0
2017/07/19 15:36:36,52.217893,6.897492,6A:76:EA:0B:0B:3E,KPN Fon,55,0,0
2017/07/19 15:36:36,52.217893,6.897492,78:C1:A7:0A:11:5B,Fra&FaWiFi,63,0,0
2017/07/19 15:36:36,52.217893,6.897492,7C:E9:D3:40:3B:BB,RIBA2,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,90:5C:44:D0:C6:16,Ziggo9F59D6C,37,0,0
2017/07/19 15:36:36,52.217893,6.897492,22:25:64:D1:0C:68,Ziggo,41,0,0
2017/07/19 15:36:36,52.217893,6.897492,84:9C:A6:E1:94:0D,Kalanderstraat10b,49,0,0
2017/07/19 15:36:36,52.217893,6.897492,20:25:64:D1:0C:67,Ziggo96310,39,0,0
2017/07/19 15:36:36,52.217893,6.897492,1C:74:0D:3C:22:2E,ZyXEL3C222D,43,0,0
2017/07/19 15:36:36,52.217893,6.897492,B0:C2:87:7A:87:B2,Ziggo02514,43,0,0
2017/07/19 15:36:36,52.217893,6.897492,74:67:F7:13:95:50,KFC-FREE,41,0,0
2017/07/19 15:36:36,52.217893,6.897492,F0:92:1C:B0:3E:CD,HP-Print-CD-Photosmart 5520,41,0,0
2017/07/19 15:36:36,52.217893,6.897492,AC:86:74:6B:8A:A2,kpnretail1,51,0,0
2017/07/19 15:36:36,52.217893,6.897492,B2:C2:87:7A:87:B3,Ziggo,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,74:26:AC:7F:41:02,VF-Guest,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,AC:64:62:6D:A7:B2,H369A6DA7B2,43,0,0
2017/07/19 15:36:36,52.217893,6.897492,74:67:F7:13:95:51,KFC-20440,41,0,0
2017/07/19 15:36:36,52.217893,6.897492,02:1A:11:FE:DB:80,HUAWEI P9,39,0,0
2017/07/19 15:36:36,52.217893,6.897492,1C:87:2C:69:86:80,Asus,53,0,0
2017/07/19 15:36:36,52.217893,6.897492,74:26:AC:6D:E4:50,VF-Corporate,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,74:26:AC:6D:E4:51,Vodafone NL Wifi,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,F0:5C:19:66:B2:C0,JdGW1ReLeSs,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,46:D9:E7:61:E6:27,CS-INT#01,37,0,0
2017/07/19 15:36:36,52.217893,6.897492,90:5C:44:9B:ED:12,ZiggoF481E69,41,0,0
2017/07/19 15:36:36,52.217893,6.897492,94:B4:0F:C1:C2:C5,43,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:B9:BE:52,Enschede_Stad_van_Nu,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:B9:BE:53,\x00\x00\x00\x00\x00\x00,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,94:B4:0F:C1:C2:C2,wifistaff,41,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:22:61:D9:B8:BE,Headless_00:22:61:d9:b8:be,49,0,0
2017/07/19 15:36:36,52.217893,6.897492,88:D2:74:82:2B:7F,H369A822B7F,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,74:26:AC:7F:41:00,VF-Corporate,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,74:26:AC:7F:41:01,Vodafone NL Wifi,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,02:CA:FE:CA:CA:40,49,0,0
2017/07/19 15:36:36,52.217893,6.897492,54:FA:3E:37:C2:F7,HZN243051973,53,0,0
2017/07/19 15:36:36,52.217893,6.897492,8A:15:04:8D:EA:90,\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
2017/07/19 15:36:36,52.217893,6.897492,8A:15:04:8D:EA:96,\x00\x00\x00\x00\x00\x00\x00,61,0,0
2017/07/19 15:36:36,52.217893,6.897492,8A:15:04:8D:EA:97,\x00\x00\x00\x00\x00\x00\x00\x00,49,0,0
2017/07/19 15:36:36,52.217893,6.897492,94:B4:0F:C1:C2:C4,wifiMD,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:B9:BE:50,eduroam,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,8A:15:04:8D:EA:98,\x00\x00\x00\x00\x00\x00\x00\x00,59,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:B9:BE:54,\x00\x00\x00\x00\x00\x00\x00,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,70:E4:22:A4:B8:30,CTX,33,0,0
2017/07/19 15:36:36,52.217893,6.897492,00:0F:7D:BC:9B:95,eduroam,47,0,0
2017/07/19 15:36:36,52.217893,6.897492,70:E4:22:A4:B8:31,nmn,39,0,0
2017/07/19 15:36:36,52.217893,6.897492,88:DC:96:46:20:F0,PA-LAN,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,C0:FF:D4:B2:D6:AE,Russ’ Network_EXT,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,54:FA:3E:5C:7C:ED,CelenoInitialAP5C7CED,37,0,0
2017/07/19 15:36:36,52.217893,6.897492,88:DC:96:4A:63:AA,Die Grenze Popup Store,43,0,0
2017/07/19 15:36:36,52.217893,6.897492,2C:C5:D3:30:B1:B8,KPN,55,0,0
2017/07/19 15:36:36,52.217893,6.897492,88:03:55:B4:30:1D,VGV7519B4301D,55,0,0
2017/07/19 15:36:36,52.217893,6.897492,94:B4:0F:C1:C2:C3,wifitda,47,0,0
```

```
2017/07/19 15:36:36,52.217893,6.897492,72:64:62:6D:A7:B3,KPN Fon,43,0,0
2017/07/19 15:36:36,52.217893,6.897492,74:26:AC:6D:E4:52,VF-Guest,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,56:D9:E7:61:E6:27,CLAUDIASTRATER,43,0,0
2017/07/19 15:36:36,52.217893,6.897492,88:03:55:CB:B8:39,KPN-VGV7519CBB839,43,0,0
2017/07/19 15:36:36,52.217893,6.897492,84:D4:7E:12:AE:C0,AT-Devices,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,B0:C2:87:7C:2D:CA,fmen,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,B2:C2:87:7C:2D:CB,Ziggo,45,0,0
2017/07/19 15:36:36,52.217893,6.897492,54:BE:F7:19:91:85,@Thuis53a,51,0,0
2017/07/19 15:36:36,52.217893,6.897492,F0:5C:19:66:B0:40,JdGW1ReLeSs,51,0,0
2017/07/19 15:36:36,52.217893,6.897492,90:5C:44:08:B9:F9,Vodafone Wifi,41,0,0
2017/07/19 15:36:36,52.217893,6.897492,5C:DC:96:E2:76:14,VGV7519E27614,61,0,0
2017/07/19 15:36:36,52.217893,6.897492,84:18:3A:34:46:D8,Ziggo,37,0,0
2017/07/19 15:36:36,52.217893,6.897492,D2:B2:C4:25:42:0D,Ziggo,39,0,0
2017/07/19 15:36:36,52.217893,6.897492,F8:04:2E:A3:90:78,UPC242836951,43,0,0
2017/07/19 15:36:36,52.217893,6.897492,9C:80:DF:D0:91:80,KPN-VGV7519D09180,47,0,0
```