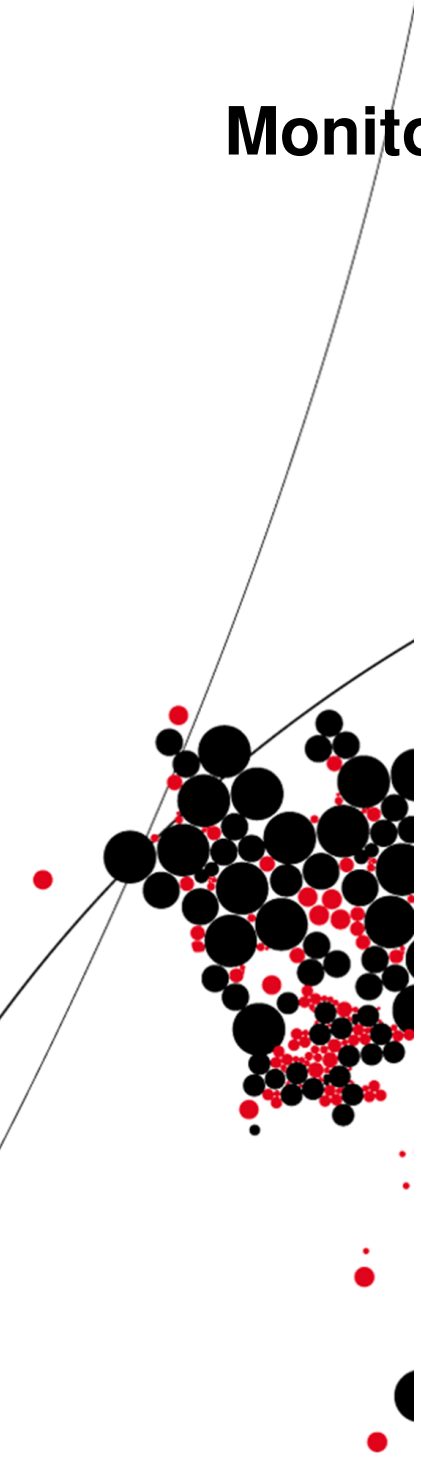
UNIVERSITY OF TWENTE.

**Faculty of Electrical Engineering,
Mathematics & Computer Science**

Monitoring crowd dynamics by passively sniffing cellular traffic

T.H. Redonet Klip

MSc Thesis
August 2020



Supervisors:
prof. dr. ir. M.R. van Steen
dr. ir. A.B.J. Kokkeler
dr. ing. N. Alachiotis

Digital Society Institute
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

Monitoring crowd dynamics assists in a multitude of scenarios. Examples are safety regulations at events, urban planning, and marketing strategies. A common way to monitor crowds is to detect smartphones by capturing Wi-Fi probe requests. However, this method produces noisy datasets. In this work, alternative ways to detect mobile phones are trying to be found. This research chose to focus on cellular networks, in particular the GSM network. GSM uses paging requests to wake up mobile phones for incoming traffic. These paging requests contain an identifier, the TMSI. The assumption is made that mobile phones can be detected, identified, and tracked through the paging requests they receive. This work conducted several experiments to validate this assumption. A Software Defined Radio (SDR) Dongle was used to sniff paging requests. Simultaneously, a mobile phone with custom firmware was used in different scenarios. It turns out that the paging channel of the GSM network does not provide enough paging requests to track mobile phones. However, more modern cellular networks like LTE, which is used for more applications than GSM, might have more potential since they also use paging requests containing a TMSI. This work also did a literature research on physical layer identification, where wireless devices are identified by the raw radio waves they emit.

Contents

List of Acronyms

AGCH	Access Granted Channel
ARFCN	Absolute Radio Frequency Channel Number
BCCH	Broadcast Control Access Channel
BTS	Base Transceiver Station
CCCH	Common Control Channel
CCH	Control Channel
CCTV	Closed-Circuit Television
CSI	Channel State Information
DCCH	Dedicated Control Channel
DSP	Digital Signal Processing
ETSI	European Telecommunications Standards Institute
FACCH	Fast Associated Control Channel
FCCH	Frequency Correction Channel
FDMA	Frequency Division Multiple Access
GSM	Global System for Mobile Communications
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
I/Q	In-Phase and Quadrature Components
LTE	Long Term Evolution
MCC	Mobile Country Code
MNC	Mobile Network Code
MSIN	Mobile Subscriber Identification Number
NFC	Near Field Communication

OSI Open Systems Interconnection

PARADIS Passive Radiometric Device Identification System

PCA Principle Component Analysis

PCH Paging Channel

RACH Random Access Channel

RSSI Received Signal Strength Indicator

SACCH Slow Associated Control Channel

SCH Synchronization Channel

SDCCH Standalone Dedicated Control Channel

SDR Software Defined Radio

TCH Traffic Channel

TCH/F Full-rate Traffic Channel

TCH/H Half-rate Traffic Channel

TDMA Time Division Multiple Access

TMSI Temporary Mobile Subscriber Identity

VLR Visitor Location Register

Chapter 1

Introduction

1.1 Motivation

Automatically measuring pedestrian dynamics is important for numerous reasons, such as safety concerns, urban planning, and marketing strategies. It covers both indoor and outdoor situations; from a rather small office building to a large shopping mall, from a train station to a citywide festival. Until now, the most common way to monitor crowd dynamics apart from CCTV systems is through Wi-Fi. This is shown, for example, by a survey [1] that reviewed over ninety crowd management systems where the majority of systems that did not require its users to use a dedicated application used Wi-Fi. However, modern mobile phones have a multitude of wireless technologies which could be used to monitor crowds. Most modern smartphones are equipped with NFC, Bluetooth, Wi-Fi, GSM, UMTS, and LTE. Cellular networks have potential to be successful in outdoor situations since they already serve mobile phones outdoors. In this research, alternative ways of identifying personal communicating devices are trying to be found. In particular, non-intrusively identifying mobile phones such that the privacy of the owner is not at stake, opposed to most existing methods which infringe upon privacy regulations.

1.2 Problem Statement

The common approach to gather data for monitoring crowd dynamics has been to capture probe requests which are sent by Wi-Fi enabled devices such as smartphones. The probe requests are captured using multiple Wi-Fi hotspots. These probe requests which are sent from Wi-Fi enabled devices are used to find nearby Wi-Fi access points. The underlying assumptions are that the MAC address in these packets is uniquely bound to the device (which is generally true) meaning that such an address can be used as an identifier, and that the movements of the device can be determined by looking at the Wi-Fi hotspots that the smartphone has been in close proximity to. Apart from the fact that this method infringes privacy regulations, it has also been shown to perform very badly in outdoor environments since the movements of smartphones seem to show erratic behaviour [2]. There have been attempts to sanitize the noisy datasets [3], but that only cures the symptoms rather than eradicate the problem which is causing the noise in the first place.

1.3 Research Questions

In order to find a new way to gather data that is suitable for monitoring crowd dynamics the following research questions are addressed to identify different subproblems.

- How to detect mobile phones?
- How to identify mobile phones?
- How to track a mobile phone?
- How to distinguish mobile phones?

How to detect mobile phones?

Different technologies which are present in modern mobile phones can be used to detect mobile phones. The probe request method using Wi-Fi is only one example. Near Field Communication (NFC), Bluetooth, Wi-Fi, and cellular networks will be considered for detecting mobile phones. Several aspects such as range and how frequently the wireless technology is used should be compared. The ease of experimenting is also different for each wireless technology. Lastly, the fact that the Wi-Fi probe request method did not perform well should also be taken into account.

How to identify mobile phones?

When a method to detect mobile phones is determined, the mobile phones need to be identified. This identifier can either be constructed or extracted from the data. For example, the Wi-Fi probe request method uses the MAC address as the identifier. When such an identifier is not available or easily accessible, an identifier needs to be constructed using other identifying factors. It is also important that this constructed identifier is consistent, meaning that it does not change each time it is constructed. This research aims to find a non-intrusive method, which makes it highly likely that the payload is not accessible when the data is encrypted. It would appear that encrypted data cannot be used to identify a device, i.e., it can be complicated to prove that data captured at time T_1 does or does not originate from the same device as the data captured at time T_0. However, wireless devices can also be identified without using the payload that is transmitted [4].

How to track a mobile phone?

Once a mobile phone is identified it needs to be monitored for an amount of time in order to determine its movements. A single detection of a mobile phone is not enough to monitor its movements. Also, an estimation of the location of the mobile phone needs to be determined. This does not need to be the exact location, but when the accuracy of the mobile phone's location increases, the granularity of the crowd monitoring system increases.

How to distinguish mobile phones?

The last subproblem is how to distinguish traffic from different mobile phones. This might seem already treated in 'how to identify mobile phones', but although closely related, they are different problems to solve. When an identifier already exists in the observed data, then the identifier is the distinguishing factor. When an identifier needs to be constructed from the data, then data from one mobile phone needs to be distinguished from data from another mobile phone to uniquely identify a mobile phone.

1.4 Organization

This thesis contains eight chapters. After Chapter 1, this chapter, follows Chapter 2 about related research, discussing a number of papers about crowd managing. One of those is a survey which reviewed over ninety crowd managing methods. Next is Chapter 3 about the Global System for Mobile Communications (GSM) network. Why this research used the GSM network is treated in Chapter 4, but Chapter 4 contains technical details where background knowledge is required in order to understand the technical details. Chapter 3 will be a brief introduction to the GSM since treating the whole protocol would be too extensive. At the end of Chapter 3 is a section about an open source GSM project that will be used in this research.
Chapter 4 will present a new method to monitor crowd dynamics. As a framework, the research questions from Section 1.3 are mentioned and how the presented method will answer the research questions.
Chapter 5 will address the practical part in this research. The experiments described in this chapter serve as a test to see if the presented method from Chapter 4 is successful. The results from these experiments will also be treated in this chapter.
Chapter 6 proposes a more extended recommendation regarding future research. Instead of focusing on one interface, network, or protocol, a suggestion is made to study the radio waves which mobile phones emit.
Chapter 7 will draw the conclusions from this research which are based on the results described in Chapter 5. It also addresses some recommendations for future work based on the results described in Chapter 5 and on the hypothesis in Chapter 6.
Lastly, Chapter 8 will address remarks on some problems that this research faced.

Chapter 2

Related Work

Draghici et al. published a survey on crowd managing in 2018 [1]. This survey provides an overview of ways to automatically sense the behaviour of a crowd using wireless technologies. The motivation for this survey is that there is still a need for high-quality data sets regarding crowd movements. Draghici et al. reviewed 93 papers in different categories. Crowd managing in urban areas, indoors, or at a large event were considered. They included papers which emphasized on different aspects: privacy, evaluation methodologies, and heterogeneity of sources. All of the reviewed papers were classified based on the following criteria: security and privacy, ease of deployment, scalability, incentives, transparency, and resource consumption. Unfortunately, no solution has been found which adequately addresses all these aspects. They have not been able to properly review accuracy due to different analysis methods and metrics that were used in the reviewed papers, making it hard to normalize accuracy. The results in the reviewed papers were also very divergent. Draghici et al. conclude that only combining app-driven and infrastructure-based solutions will come to decent solutions for sensing crowds.

Monitoring crowds by looking at mobile phones is only one aspect of managing crowds. Acting on observations is another aspect. In 2016 Wijermans et al. presented Incrowd, a decision-support framework for crowd management [5]. It supports and provides observing, interpreting, predicting, and decision making. To demonstrate Incrowd, they used the World Living Statues festival in the city of Arnhem in The Netherlands. This event attracted over 300.000 visitors. The event was monitored using about 80 cameras, 50 Wi-Fi hotspots to detect mobile phones, and contributions from security guards. The framework is intended to function as an architecture for systematic decision making in crowd management as well as a model development framework. The framework consists of four fragments. First, the crowd-interaction system, which is the interface between the crowd and the crowd managers. This is through sensors and actuators. Second, the data mining system, which interprets the raw data from the sensors. Third, the predicting system, which predicts the state of the crowd. Last, the decision-making system, which decides whether a crowd needs intervention through the actuators.

Wijermans et al. argue that existing crowd managing strategies rely on models which are not systematically analysed. The trend in crowd managing is mostly preparation for potential and common scenarios rather than real-time situation reports. Multiple disciplines should be integrated in crowd managing methods according to Wijermans et al. In particular, the connection between computer science and social sciences have lots of room for improvement. Incrowd is a high-level architecture for decision-support in crowd management, which also clearly addresses shortcomings in the research in crowd managing techniques.

One of the entry points for this research was the research presented by Chilipirea et al. in 2016 [3]. This study approaches monitoring crowds by detecting smartphones using Wi-Fi. Wi-Fi scanners are strategically placed in the area which needs to be monitored. These scanners are capable of sniffing MAC addresses from Wi-Fi probe requests. The MAC addresses are hashed so the privacy of the owners of the smartphones is preserved. The resulting data set is a multitude of lists of timestamped detections of devices, each Wi-Fi scanner with its own list. This should allow tracking of devices since the location of the Wi-Fi scanners is known. This experiment was conducted on a three day festival visited by around 130.000 people.
Chilipirea et al. anticipated on a highly noisy dataset so the focus of this study lies on the systematic extraction of useful information from the data set in order to enable effective crowd monitoring. The noise was mainly caused by devices using short-living network addresses, devices being in range of multiple scanners, but most importantly the relatively low frequency of detections of devices. Chilipirea et al. propose three methods to sanitize noisy data sets: removing low-quality detections, averaging detections, and eliminating repetitive behaviour. Low-quality detections are defined by the Received Signal Strength Indicator (RSSI). Detections with an RSSI value lower than a defined threshold are discarded. Detections from one device are averaged over a defined time window, so each time window is left with one detection per device. The *cycle removal method* was specifically designed to remove a phenomenon which arose in the datasets, where some devices appeared to move erratically back and forth or in circles. To evaluate the performance two metrics were used. One being the entropy, which defines how much noise there is, and the other being the dissimilarity between the original data set and the modified data set.
Chilipirea et al. came to the conclusion that each solution has its own advantages and the choice for a solution depends on the application. But, the cycle removal method shows to be successful to smooth a fuzzy path or to identify a static device. The disadvantage of the study that more studies suffer from is that their dataset lacked a 'ground truth' to validate their evaluation.

Chapter 3

The GSM Network

The GSM network is the second generation mobile network used by mobile devices, hence the common reference 2G. It was first deployed in 1991 by the European Telecommunications Standards Institute (ETSI). It operates on the 900MHz and 1800Mhz bands.

GSM uses Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) to allow multiple devices simultaneously. TDMA allows multiple devices to use the same frequency. It assigns time slots to each mobile device to make use of the connection in turn. To make FDMA work, the GSM band is divided in many sub frequencies 200kHz separated from each other. GSM is a bidirectional channel which means that downlink and uplink happen simultaneously. This is why the downlink and uplink frequency have a relative offset of 45MHz. For example, a sub frequency with a downlink channel on 955.4MHz has an uplink channel on 910.4.
Each of those frequencies is denoted by an Absolute Radio Frequency Channel Number (ARFCN) and is defined by $f_{Downlink} = 935 + 0.2ARFCN$. For example, the carrier frequency 955.4Mhz has an ARFCN of 102.
The GSM interface has 3 layers and is called the air interface or Um interface. It is called the Um interface because it is the mobile variant of the U interface from Integrated Services Digital Networks (ISDNs), where 'm' stands for mobile. The three Um layers represent the physical layer, the datalink layer and the network layer in the OSI model [6]. In the GSM protocol encryption is implemented in the first layer, so most data from the datalink layer and the network layer is out of reach. Hence, this research will mainly focus on the physical layer since data from the other layers is out of reach.
Not all TDMA timeslots contain the users speech or data, but they have different tasks as well. GSM has several logical channels that are responsible for maintaining and managing the connection with multiple mobile phones. These channels are elaborated upon on the next page.

3.1 Channels

The GSM's logical channel can be divided in Traffic Channels (TCHs) and Control Channels (CCHs). These channels can be further divided in 2 and 9 different channels, respectively. The diagram in Figure 3.1 helps to illustrate the hierarchy.

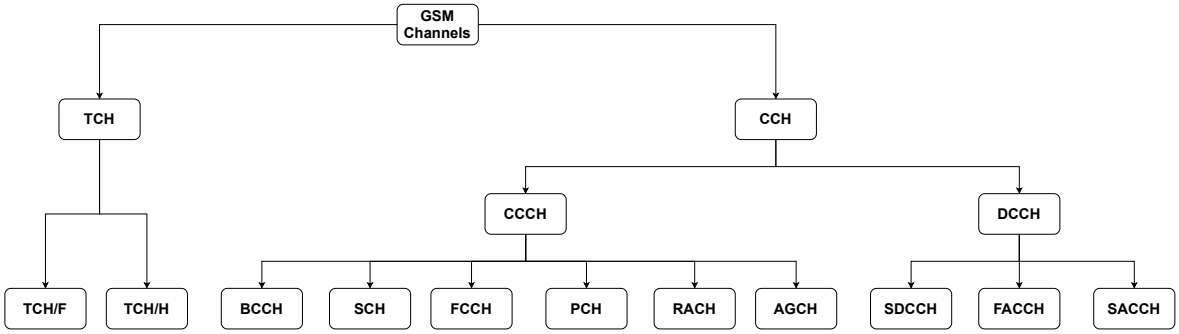


Figure 3.1: This figure is an illustration of the hierarchy of GSM's logical channels. There are relatively many CCHs, each with a dedicated task. Every channel is assigned one or more time slots in the TDMA scheme.

Traffic Channels (TCHs)

This channel carries, unsurprisingly, the actual traffic. This channel can either be a Full-rate Traffic Channel (TCH/F) or a Half-rate Traffic Channel (TCH/H). The latter is used, so that one time slot can be shared with another user. This will decrease the throughput though.

Control Channels (CCHs)

Control Channels (CCHs) are managing channels. They are reserved for all signaling data so that the protocol actually works. All CCHs have a specific function contributing to this. The Broadcast Control Access Channel (BCCH) sends data with information about the Base Transceiver Station (BTS), such as the identifier of the BTS and the mobile operator that the BTS belongs to. This data is meant for all devices in reach of the BTS, including devices that are not attached to the BTS. Mobile devices determine the quality of the reception of the BTS from the BCCH.

The Synchronization Channel (SCH) and Frequency Correction Channel (FCCH) make sure that the mobile phone can set its timing right to make use of the network. Since GSM uses TDMA, timing is essential and data could clash if the timing is not correct. When connecting to the network, the mobile phone waits for a frame on the SCH to set its timing right. The frames on the FCCH are used to tune the mobile phone's oscillator.

The PCH is a channel that it used by the BTS to 'awake' a mobile phone so that the mobile phone requests a channel to send and receive data on. This happens when a mobile phone

receives incoming traffic. When the mobile phone is idle it only needs to listen on this channel for paging requests. Since this channel is not active very much this is energy efficient. When a mobile phone receives a page from the BTS it will initiate a connection by requesting a connection on the Random Access Channel (RACH). The BTS will then respond on the Access Granted Channel (AGCH) to assign a slot on the TCH.

Previously discussed Control Channels (CCHs) are all Common Control Channels (CCCHs). The last three channels to be treated are the Dedicated Control Channels (DCCHs). These channels focus on managing individual connections rather than managing the network. These can be used to setup a call or send SMS messages. Also registering and ciphering setup happen on these channels. All four tasks happen on the Standalone Dedicated Control Channel (SDCCH). Every SDCCH has an associated Slow Associated Control Channel (SACCH) to carry general information. Think of instructions for timing or transmitting power. The SACCH can also be used to send or receive SMS messages on during a call. Not only every SDCCH but also every TCH has a dedicated SACCH. The Fast Associated Control Channel (FACCH) is used to carry urgent information. Every TCH has an associated FACCH. The FACCH can claim a time slot previously occupied by the TCH by setting the 'stealing bit'.

3.2 Frames

GSM uses TDMA to allow multiple users on the same physical channel. It also uses TDMA to allow multiple logical channels on the same physical channel. This means that multiple users can use multiple logical channels on the same physical channel. A physical channel is divided into 8 time slots, or bursts, which forms one TDMA frame. Such a frame has a duration of approximately 4,615ms. This yields that a mobile device only needs to be active on this physical channel for 576,2μs. Remember that there is a physical downlink channel and a physical uplink channel. Each user is assigned the same time slot on the down and uplink channel. However, the timeslots on the uplink channel have an offset of 3 with respect to the downlink channel so that the mobile device does not have to transmit and receive at the same time, which is a lot less complicated.
Multiple frames make up a multiframe. There are two different multiframes. 26 frames make up a traffic multiframe and 51 frames make up a control multiframe. Multiple multiframes compose a superframe. Just like multiframes, there are two different superframes. *51* traffic multiframes make a traffic superframe and *26* control multiframes make a control superframe. Note how each superframe has an equal duration where different multiframes have different durations. This is because all superframes last 26 x 51 x 4,615ms = 6,119s. Lastly, 2048 superframes compose one hyperframe which lasts approximately three and a half hours. Each frame in a hyperframe has a frame number which is used, for example, as an encryption parameter.

Counterintuivitively, not all frames in a traffic multiframes are reserved for the TCH. Each 13^{th} frame is occupied by the SACCH and the last frame is left idle. This means 12 frames for the TCH, one for the SACCH, 12 frames TCH and one left idle, in that order. The composition of control multiframes is a bit more complex and not always the same. The exact composition of a control superframe may depend on the configuration of the BTS. An example of a composition can be found in Figure 3.2 below.

Downlink			
0	FCCH	26	CCCH
1	SCH	27	
2	BCCH	28	
3		29	
4		30	FCCH
5		31	SCH
6	CCCH	32	CCCH
7		33	
8		34	
9		35	
10	FCCH	36	CCCH
11	SCH	37	
12	CCCH	38	
13		39	
14		40	FCCH
15		41	SCH
16	CCCH	42	CCCH
17		43	
18		44	
19		45	
20	FCCH	46	CCCH
21	SCH	47	
22	CCCH	48	
23		49	
24		50	Idle
25			

Uplink			
0	RACH	26	RACH
1	RACH	27	RACH
2	RACH	28	RACH
3	RACH	29	RACH
4	RACH	30	RACH
5	RACH	31	RACH
6	RACH	32	RACH
7	RACH	33	RACH
8	RACH	34	RACH
9	RACH	35	RACH
10	RACH	36	RACH
11	RACH	37	RACH
12	RACH	38	RACH
13	RACH	39	RACH
14	RACH	40	RACH
15	RACH	41	RACH
16	RACH	42	RACH
17	RACH	43	RACH
18	RACH	44	RACH
19	RACH	45	RACH
20	RACH	46	RACH
21	RACH	47	RACH
22	RACH	48	RACH
23	RACH	49	RACH
24	RACH	50	RACH
25	RACH		

Figure 3.2: Example of a composition of a control multiframe. On the downlink channel there is a FCCH and a SCH frame every 10th frame so that mobile phones can adjust their timing and oscillators. The first four frames which are not FCCH or SCH are allocated for the BCCH. The rest of the frames are allocated for the rest of the CCCHs, which are the PCH and AGCH. This structure is called the BCCH/CCCH 51-frame structure.

3.3 Coverage

One of the physical elements of the network are the GSM masts which are called Base Transceiver Station (BTS), or simply base station. Each BTS is equipped with one or multiple transceivers. The transceivers are the actual antennae and provide the transmission of the radio signals.
BTSs have their transceivers set up in sets of three in a mast. Each transceiver points in a different direction with equal angles between one another. This implies that each transceiver has a range in an isosceles diamond shape with an angle of 120° and a finite height. Formally, these areas are called sectors and each mast has three of them, one for each transceiver. Three of those sectors make a cell. Intuitively, one might think that those three are the surrounding sectors around a tower but that is not true. One cell consists of three sectors from three different BTSs with one BTS in each corner. The cell then forms a hexagon. The principle is exemplified in Figure 3.3 below.

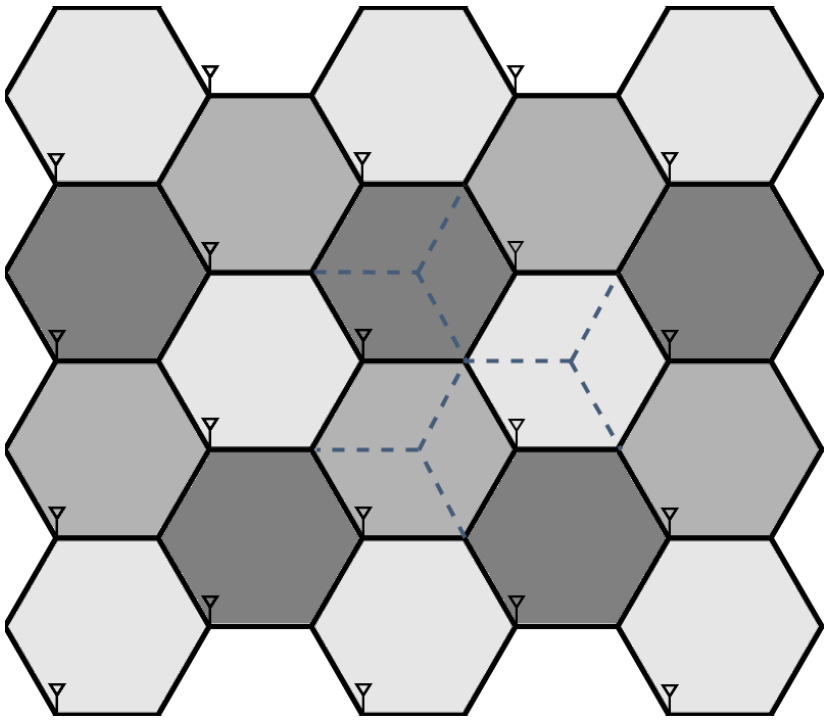


Figure 3.3: This figure illustrates the theoretical grid in which BTSs are positioned. The antenna icon represents a BTS. Each grey area is a GSM cell. As can be seen, each cell is made up of three sectors from three different BTSs.

In practice the BTSs cannot be placed in a perfect grid with equal distances from each other. An example can be seen in Figure 3.4. What can also be seen from the figure is that the shape of a cell can be very irregular. The borders of these cells are not very precise since reception is very dependent on ambient factors such as buildings, weather, or people.

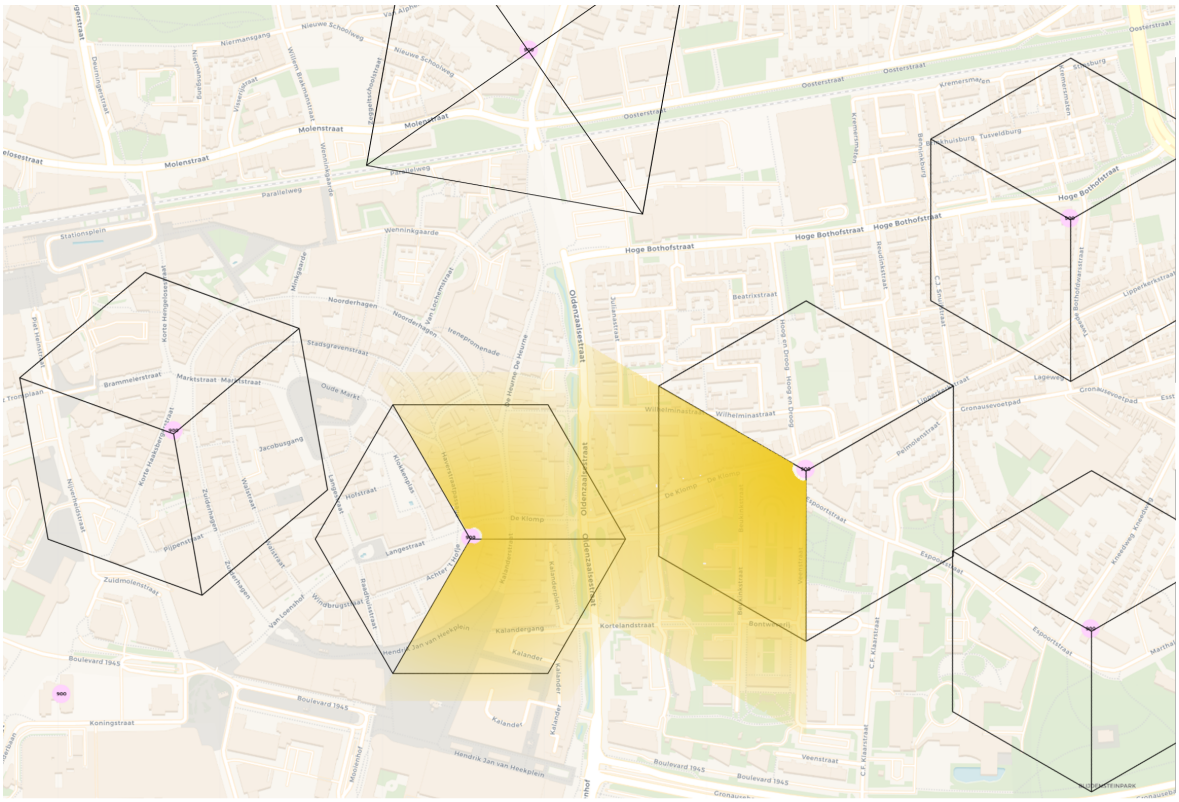


Figure 3.4: Real world example of the positions of the BTSs in a city. In this example the city centre of Enschede can be seen with the locations of the BTSs from one mobile operator for the GSM900 band. Each pink dot is a BTSs and each of the diamond shapes surrounding the pink dots represent the operational area of the transceivers. The orange shape is one cell which consist of three transceivers of two BTSs.

3.4 Identifiers

IMSI

The International Mobile Subscriber Identity (IMSI) [7] is a permanent identifier for every user in a mobile network. It is usually embedded in the SIM card since the SIM card is considered the user of the mobile network, opposed to the International Mobile Equipment Identity (IMEI) which is bound to the physical device. The SIM card uses the IMSI to identify itself to the BTS and does a cryptographic operation to authenticate itself to the BTS. Since the IMSI is a permanent identifier, it is susceptible to tracking and thus privacy infringement. Therefore, it will only be used when strictly necessary. BTSs keep IMSIs in their Visitor Location Register (VLR). More is discussed the next section.

The IMSI consists of three values which add up to a total of 64 bits. It starts with a Mobile Country Code (MCC), which is quite self-explanatory. The first digit of the MCC represents the continent, the other two digits the country. Next is the Mobile Network Code (MNC). Each network operator is assigned a unique MNC in each country. Combined, the MCC and MNC uniquely identify every network operator worldwide. The Mobile Subscriber Identification Number (MSIN) makes up the rest of the IMSI and consists of 9 or 10 digits. This depends on whether the MNC consists of 2 or 3 digits, but this is country specific. The MSIN is the actual SIM card identifying part of the IMSI. Table 3.1 below demonstrates the composition of an IMSI for clarification.

Table 3.1: This example IMSI, 204 20 1234567890, has MCC 204 and MNC 20 which belong to The Netherlands and T-Mobile, respectively.

MCC			MNC		MSIN									
2	0	4	2	0	1	2	3	4	5	6	7	8	9	0

TMSI

The Temporary Mobile Subscriber Identity (TMSI) [7] is an identifier which is randomly generated by the BTS and kept in the VLR. It is meant to replace the IMSI to preserve the users privacy. An IMSI is a permanent identifier bound to a SIM card. Since identifiers, both TMSIs and IMSIs, are sent over the air without any encryption IMSIs are prone to tracking. TMSIs solve this problem. They are assigned to each subscriber and are updated each time a subscriber switches to a different BTS. TMSIs are used by the BTS for paging requests, and by mobile phones for paging responses. Paging is the act of awaking a mobile phone for incoming traffic by the BTS. A TMSI consists of 32 bits and is often represented in eight hexadecimal digits. Since it is only of local importance, the BTS is responsible for distributing TMSIs and keeping track of mappings from IMSI to TMSI. The TMSI allocation and reallocation procedures are described in the protocol [7], but it depends on the configuration of the BTS how often and on which conditions TMSIs are reallocated.

3.5 OsmocomBB

Although the GSM documentation is publicly available, these documents only give a theoretical description of the protocol. Most embedded systems implementing the protocol, for example in mobile phones, are not open source. A lot of mobile phones have android as its operating system which is open source, which might cause a misconception that the implementation of the network protocols is also open source. However, mobile phones have a dedicated processor called the *baseband* to implement mobile network protocols. The baseband has a driver to use it and might have a debug interface to extract more information, but due to the complexity of these chips it is not efficient for analysis. The baseband is still a dedicated processor with its own firmware.
At some point the documentation of a baseband chipset, published in the year 2000, was leaked. The documentation was a manual for the registers of the Calypso chipset manufactured by Texas Instruments. The Calypso chipset was used in, for example, Motorola cellphones in the mid 2000s. These manuals seemed to be very useful not only for reverse engineering but also for a full reimplementation of the firmware of the Calypso chipset. This is realised in the OsmocomBB project [8], which calls itself an open source GSM baseband implementation. OsmocomBB stands for Open Source MObile COMmunications BaseBand. OsmocomBB is one of Osmocom's projects and its development started in 2010. Its goal was to give a better understanding of, and practical insights in the GSM network. Its software gives full control over the hardware and software which is very useful for analytical purposes. For example, a very useful feature is that GSM traffic can be encapsulated in UDP/TCP packets and analysed in a packet sniffing tool like Wireshark.

Researches involving OsmocomBB

The Osmocom project [9] has been around quite some time and the GSM network even longer. At the time of this research recent studies involving OsmocomBB are hard to find. Nevertheless, through the years there have been studies of similar extent as this research where OsmocomBB played a significant role.
Van den Broek [10] published a research in 2010 which used OsmocomBB to give insight in the GSM protocol for those who want to experiment with GSM. This kind of work is especially relevant because the GSM world is so closed despite its documentation being publicly disclosed. This work also made a slight emphasize on the security flaws of GSM. This was all shown by exposing network traffic using OsmocomBB.
With the help of the previously mentioned research [10], Bosma [11] tried to exploit one of these security flaws in practice. They targeted the imperfect encryption algorithm called A5/1 which GSM uses. They used the OsmocomBB for reconnaissance of the GSM network before launching an attack on the encryption. Although not implemented for security reasons, the Dutch networks all use frequency hopping which prevented them from successfully performing the attack.

Another attempt to put theoretical attacks on the GSM protocol into practice was presented in [12]. This work actually used OsmocomBB for active security analysis. Here a mobile phone with the OsmocomBB software is used for both passive and active attacks. It is for eavesdropping where encryption is cracked offline, and a Denial-of-Service attack. Only the latter succeeded.

The researches mentioned above all focused in some way on the security aspect of the GSM protocol. In that manner they differ from this research since they looked for things to break or exploit in the network. OsmocomBB might be used slightly differently for this reason. This research has no interest in abusing the network since it might defeat the purpose of maintaining privacy of the users of the network. Above that, it is probably illegal given the purpose of this research.
In this research and all works mentioned above, OsmocomBB contributed the most to a deeper and more practical understanding of the GSM network rather than the tremendous amount of dense documentation. Some measurements and experiments were only possible thanks to this software.

Chapter 4

Monitoring crowd dynamics by passively sniffing cellular networks

4.1 Method

This chapter proposes a method to monitor crowd dynamics using the GSM network. Each GSM network consists of cells and each cell has a PCH. The PCH is used to 'awake' mobile phones when the BTS requires a connection for incoming traffic. Connections are terminated after all traffic is handled. GSM is not a packet switched network but a circuit switched network. This means that a new connection needs to be set up for every stream of traffic. More specifically, the BTS needs to assign a free time slot to the subscriber. When there is incoming traffic, the BTS pages the subscriber and it will respond with a paging response. Then, the BTS responds with a channel assignment. The paging request needs an identifier, the TMSI, so that the right subscriber responds to the request. The TMSI is uniquely bound to one device, meaning that it can be used to identify mobile phones. If all different TMSIs are captured by sniffing the PCH, the amount of mobile phones in the cell can be determined, provided that all phones are receiving paging requests. The experiments of this research will give an answer to the question whether phones are receiving (enough) paging requests. If this method can consistently determine whether a subscriber is present in a cell, then movements in and out of a cell can also be detected.
The research questions of this research were presented in Section 1.3. The following subsections will be a more elaborate explanation of the presented method using the research questions as guidelines. The presented method will be tested in Chapter 5 and Section 5.4 will evaluate whether the research questions are answered to a satisfactory level.

4.1.1 How to detect mobile phones

This subsection will consider how to detect mobile phones. As mentioned before, mobile phones are equipped with several wireless technologies. Each technology has different characteristics which may or may not be an influencing factor when used for crowd monitoring. This subsection will look at characteristics of Bluetooth, Wi-Fi and cellular networks. At the time of this research, the 5G network is in full development but has not been rolled out yet. For that reason 5G will be left out of the scope of this research. NFC will be left out of scope as well. As the name implies, NFC is designed to operate on small distances (around 10 centimeters) which is highly impractical.

. The features that will be compared are:

- Range
- Manageability
- Accessibility
- Availability
- Performance

The range refers to the maximum distance between the sensing device, like a dongle, router, or antenna, and a mobile phone. The manageability refers to the ease of experimenting with the wireless technology. The accessibility refers to how easy it is to extract information. The availability refers to how often the technology is available to extract information. The performance refers to how well it performed in previous researches regarding monitoring crowd dynamics.

Bluetooth
Most mobile phones are equipped with Bluetooth class 2, which has a maximum power of 2.5mW which translates to a range of about 10 meters with no obstructions [13]. Some scenarios can be quite challenging with this range. For example, environments where it is very difficult to place a measurement device every 20 meters, like open outdoor areas. Sensing devices could be placed underground. Although this solves the practical problem, it is expensive. Having more sensors provides more precision since the area that one sensing device can cover is fairly small. Due to its short range, Bluetooth allows to reveal when a mobile phone is nearby, so with multiple sensors a rough location of every Bluetooth enabled phone can be estimated. However, the costs will increase drastically when increasing the area to be monitored. Not all devices have Bluetooth enabled which might also be an obstructing factor. Since Bluetooth has a short range it does not require a transmitting license. This makes Bluetooth easy to set up and play with. The protocol and a lot of hardware are open source which is convenient.

Wi-Fi

Wi-Fi has an indoor range of about 10 meters. Some modern access points claim to have a range of 150 meters in direct line of sight. However, it has been shown that range is highly inconsistent, especially outdoors. Chilipirea concluded that Wi-Fi does not perform well in outdoor situations [2]. Equivalent to Bluetooth, Wi-Fi does not require a transmitting license. However, Wi-Fi devices need to meet certain requirements to operate without a license. These requirements are set by the Wi-Fi Alliance [14]. A Wi-Fi network is simple to set up for the same reasons as Bluetooth. Like Bluetooth, there is a broad market for open source consumer hardware.

Cellular Networks

Unlike Wi-Fi and Bluetooth, it can be assumed that every mobile phone has an active cellular connection. This is a very favourable feature. Cellular networks have a range of more than ten kilometers. The exact range however, is harder to determine. Buildings, people, and the density of objects which obstruct a direct line of sight have a great influence. Although cellular networks have a potential range of 30km, the BTSs in urban areas are commonly spaced only a few hundred meters apart, which can be seen in Figure 3.3, but these characteristics might differ per cellular network. At the time of this research the three most used cellular networks in The Netherlands are GSM, UMTS and LTE which respectively correspond to 2G, 3G, and 4G.

With respect to Bluetooth and Wi-Fi, cellular networks are less accessible. There is no consumer hardware to deploy a cellular network. Hardware to sniff cellular networks is scarce. While it is relatively easy to setup and manage a Wi-Fi or Bluetooth network, cellular networks are usually not meant to be managed by other instances than mobile network operators. This is the reason why cellular networks are harder to experiment with. Protocol descriptions might be open source but hardware documentation is not. The lack of practical knowledge can obstruct easy experimenting.

However, as mentioned in Section 3.5, there is a community driven software project which emerged from leaked documents of GSM hardware. This makes hands-on knowledge accessible for those who want to study cellular networks.

Comparison
Previously in this section, three wireless technologies are discussed with some of their characteristics. An overview can be found in Table 4.1.

Table 4.1: Comparison between Bluetooth, Wi-Fi, and Cellular networks.

	Bluetooth	Wi-Fi	Cellular Networks
Range	10m	10m to 150m	up to 30km
Manageability	Good	Good	Moderate
Accessibility of information	Good	Good	Moderate
Availability	Moderate	Good	Excellent
Performance in previous researches	Moderate [15]	Bad	N/A

Although Bluetooth has potential, working with a wireless connection with more range is more practical. Wi-Fi is not a bad option looking at the characteristics and the popularity in previous studies [1], despite the drawbacks. The previous section mentioned that experimenting with cellular networks is not easy compared to Bluetooth and Wi-Fi due to the lack of open source implementations of cellular networks. However, there are existing methods to analyse GSM traffic with very simple equipment. This makes experimenting more accessible. Considering the fact that Wi-Fi performs badly outdoors, this research aims to find an alternative method, and the possibility to experiment with GSM using simple equipment, this research will focus on the GSM network.

4.1.2 How to identify and distinguish mobile phones

Mobile phones are not directly detected with the method presented in this research. GSM is the medium which will be used. The GSM traffic coming from the phone is not observed, but the traffic that is sent to the phone. A simple dongle is used to sniff one of the frequencies which GSM uses. Specifically, the PCH is sniffed. More about this simple dongle is described in Section 5.1.2. Which frequency is used by which BTS and the location of the BTS is publicly disclosed [16] so it is easy to determine where the captured signals are coming from. When signals from a BTS are captured, an active device should be nearby.
Not all information which can be captured from the GSM network might be interesting. Information about which phone is assigned to which time slot is not available since that information is encrypted. But as mentioned before, a mobile phone needs to be woken up since it does not have a constant active connection. In fact, the time slot is assigned to a different phone when the connection is terminated. So only the traffic on the PCH is considered interesting. Those paging requests contain a TMSI, the identifier that the mobile phone listens to. TMSIs are directly mapped to mobile phones. This means that when different TMSIs are detected, an equal amount of mobile phones in the cell are detected.

4.1.3 How to track mobile phones

To enable monitoring crowd dynamics, movement needs to be detected. The ideal way to achieve this is to have a live location of all mobile phones in the area that needs to be monitored. However, such an ideal situation is hard to achieve. A less ideal situation may also suffice. There are roughly said two parameters that need to be taken into account.

The first is how accurate the location of the phone is. A coordinate or precise location can change into an area or circle where a mobile phone is located. The accuracy of the crowd monitoring system, or granularity, will depend on how big this area is. With the method presented in this research, the area will be one GSM cell. If a mobile phone is detected and identified by its TMSI, then the cell in which the mobile phone is located is determined. The TMSI will be its identity as long as it remains in the cell. It is assumed that when a new TMSI appears, then a new mobile phone has entered the cell. Vice versa, when a TMSI stops appearing, then a mobile phone has left the cell. If multiple cells are monitored, then movements can be detected by looking at each cell at how many phones enter and exit the cell. The sum of all phones should remain the same, so an estimate can be made in which direction a mobile phone is moving.

The second parameter to take into account is the frequency of which a phone can be detected. If phones cannot be detected frequently enough, then the number of mobile phones will be too inaccurate. If mobile phones are not detected because of the lack of traffic to the mobile phones, the performance decreases drastically. This parameter is the frequency of paging requests for one phone. The experiments discussed in Chapter 5 will determine whether the frequency of paging requests is sufficient.

Chapter 5

Experiments

5.1 Measurement Setup

5.1.1 The Motorola C139 running OsmocomBB

A Motorola C139 mobile phone which is equipped with the Calypso chipset is connected to the OsmocomBB software. A binary is loaded into memory through a serial cable and sets the stack pointer to the corresponding location in memory. This can be done by interacting with the mobile phone at the boot sequence to trigger its debug mode. The loaded binary acts as a proxy and enables full control of the device through the serial cable.
OsmocomBB comes with a program, written in C, called *mobile*. *Mobile* implements the behaviour of a regular mobile phone. The advantage of this setup is that experiments with an emulation of a regular phone can be run with access to all interesting parameters and identifiers. The TMSI and ARFCN in particular are interesting in this experiment.
Mobile is also able to encapsulate GSM traffic in UDP/TCP packets and send these packets to a user-defined IP address. This allows easy analysis in packet sniffing software like Wireshark. A block diagram of the complete measurement setup can be found in Figure 5.1 and a picture of the Motorola C139 with the serial cable can be found in Figure 5.2a.

5.1.2 Sniffing the PCH with Software Defined Radio (SDR)

Alongside the Motorola C139 is a cheap DVB-T dongle based on the RealTek RTL2832U chipset [17]. DVB-T is short for Digital Video Broadcasting - Terrestrial. This mass produced hardware is meant to receive and demodulate DVB-T signals and send MPEG-2-TS, a video stream, over USB. However, this chipset provides FM, DAB, and DAB+ support for Windows where the demodulation is done by Windows drivers. Exploiting this feature enables the transmission of raw In-Phase and Quadrature Components (I/Q). This makes Software Defined Radio (SDR) with the RTL2832U chipset possible.
Out of many community-driven software projects, the package gr-gsm by Osmocom [18] is used in this research. This package is built with GNU Radio, which is a development toolkit for building SDRs. A block diagram of the complete measurement setup can be found in Figure 5.1 and a picture of the DVB-T dongle with the antenna can be found in Figure 5.2b.

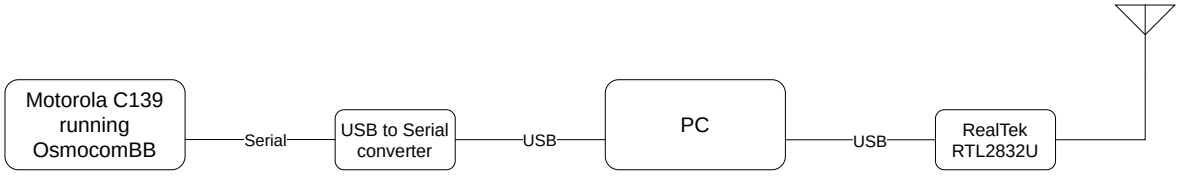


Figure 5.1: A block diagram of the setup that was used to take the measurements. The host PC used *mobile* to interact with the Motorola C138. Simultaneously, it uses a simple script built with the GNURadio library that sniffs TMSIs using the RealTek RTL 2832U . The host PC also runs Wireshark to capture the encapsulated GSM traffic coming from *mobile*.

(a) The Motorola C139 running the OsmocomBB firmware.

(b) A DVB-T dongle equipped with the RealTek RTL2832U chipset.

Figure 5.2: A picture of the setup that was used to take the measurements.

5.2 Measurements

Experiments are conducted with both setups used simultaneously. The mobile phone with the OsmocomBB firmware running will act like a proxy. *Mobile*, which acts like a normal mobile phone, will run on the host PC. Traffic from and to *mobile* will be captured, as well as the PCH, in multiple measurements with different scenarios. The scenarios are:

- Receive a call every 60 seconds, five times.
- Switch the phone on and off to observe the attaching and detaching process.
- Have the phone switched on and idle for five minutes.

In the meantime, all incoming and outgoing GSM traffic and parameters, like the TMSI and the ARFCN, are captured. This information will be compared with the measurement from the PCH later.
Next to the mobile phone will be the DVB-T dongle which captures the paging requests on the same ARFCN that *mobile* is connected to. These detections of paging requests are timestamped so a timeline can be constructed later. The output is a set of detections where each detection consists of a TMSI and an epoch timestamp. Note that this measurement will always be on the same ARFCN that *mobile* is connected to, where an ARFCN is directly mapped to a frequency.

The behaviour of the mobile phone will be visible when the captured data sent by *mobile* is analysed in Wireshark. Using the mobile phones TMSI, its behaviour can be mapped onto the Paging Channel (PCH). This will give insight whether the presented method will be able to detect, identify, distinguish, and track mobile phones. If all four aforementioned succeed the method can be assumed successful. This will be determined in Section 5.4 and in Chapter 7.

5.3 Results

In this section the results of the experiments are presented. The diagrams generated with Wireshark [19] using the captured data from *mobile* can be found in Figures 5.3, 5.5, 5.7, and 5.9. These figures contain information about all GSM traffic which *mobile* sends and receives. All traffic also includes information which is sent over the broadcast channel, paging requests for other subscribers, and channel assignments for other subscribers. This is all traffic which the mobile phone receives, but not necessarily needs. This type of traffic is marked blue in the diagrams. All incoming traffic which *is* intended for the mobile phone as well as the outgoing traffic is marked a different colour in the diagrams. On the horizontal axis of Figures 5.3, 5.5, 5.7, and 5.9 is the time in seconds and on the vertical axis is the amount of incoming and outgoing packets per second. Hence, if the value is not null there is incoming or outgoing traffic in that particular second. Paging requests for *mobile* are marked red or orange. Other GSM traffic from or to *mobile* is marked green. This is a standard diagram provided by Wireshark. Paging requests for *mobile* and other GSM traffic from or to *mobile* are highlighted with coloured bars. The width of the bars have no meaning.

The captures of the PCH with the DVB-T dongle can be found in the diagrams in Figures 5.4, 5.6, 5.8, and 5.10. On the horizontal axis of Figures 5.4, 5.6, 5.8, and 5.10 is the the time in seconds and on the vertical axis are the unique TMSIs. Each unique TMSI has been labeled with a number in chronological order, hence the diagonal line in new appearances. Since the TMSI of the mobile phone can be discovered with the OsmocomBB software, the paging requests for the indicated TMSI, the TMSI which is assigned to *mobile*, are marked red or orange. If no detections are marked red, then no paging request forthe TMSI which is assigned to *mobile* are detected. Paging requests for *mobile* are highlighted with coloured bars. The width of the bars have no meaning.

Receive a call every 60 seconds, five times

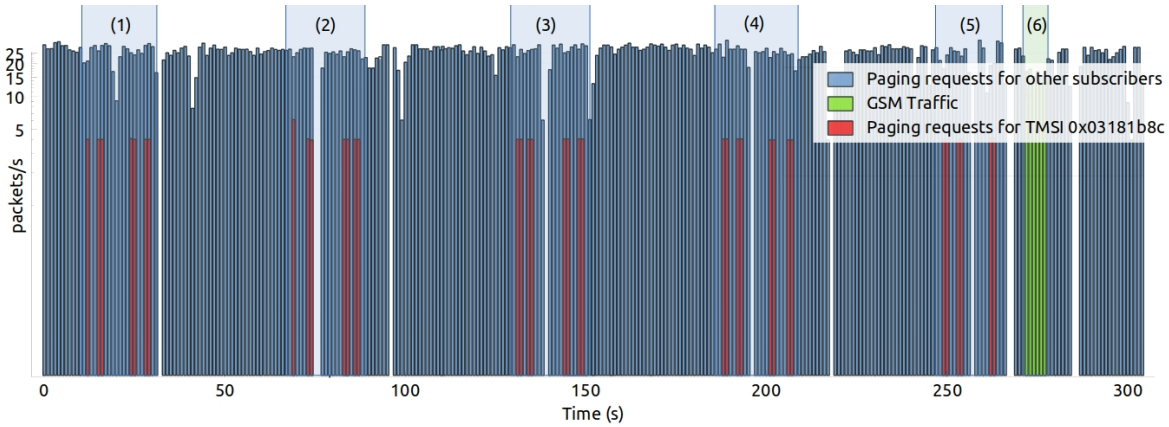


Figure 5.3: Capture from *mobile*.

The measurement from Figure 5.3 has a length of five minutes with an incoming call every minute. *Mobile* was assigned the TMSI 0x03181b8c. The paging requests for *mobile* for the 5 incoming calls are marked red. These can be found in the coloured bars marked 1 to 5. Note that the width of the bars have no meaning. The GSM traffic in the 6^{th} bar is an incoming text message from the mobile operator about a missed call. The Y-axis is on a logarithmic scale for better visibility of the paging requests in the figure.

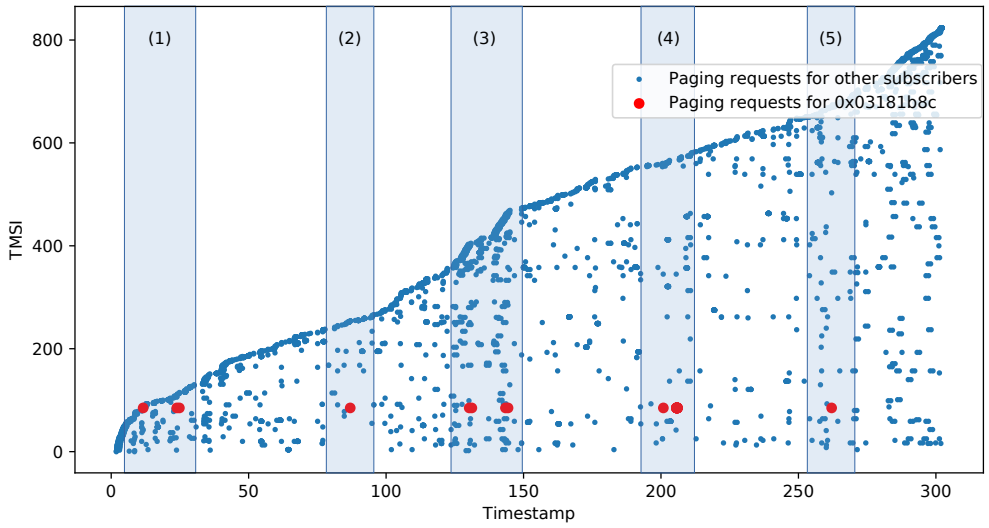


Figure 5.4: Capture from the PCH.

The measurement from Figure 5.4 was taken simultaneously with the measurement from Figure 5.3. The paging requests for TMSI 0x03181b8c are marked red. Although not all paging requests for *mobile* are detected, they match the paging requests from the 5 calls by comparing Figure 5.3 and Figure 5.4.

Unfortunately, the OsmocomBB software is very fragile and unstable. It seems that most of the time *mobile* does not respond to paging requests correctly, and therefore is unable to setup a connection. Sporadically, it successfully establishes a connection for either an incoming call or an incoming text message. However, *mobile* crashes after a connection for an incoming call is successfully established. The debug interface indicates the error as a Digital Signal Processing (DSP) error.
The following diagrams in Figures 5.5 and 5.6 show a measurement of five minutes with an incoming call every minute where *mobile* established a connection on the last call and crashed. It also received a text message after the second call. Notable is that the mobile phone is assigned a new TMSI after receiving this text message. This is probably for security concerns since text message are sent unencrypted over the SDCCH.

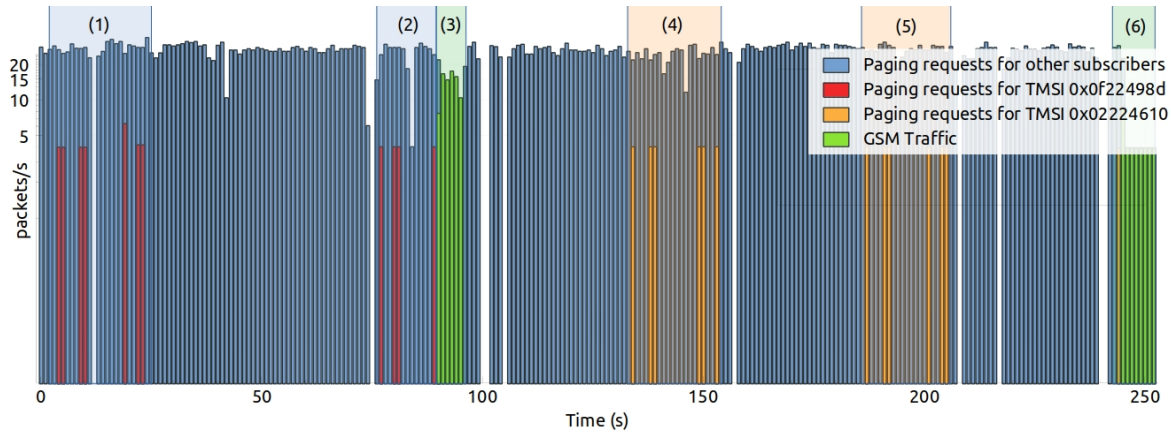


Figure 5.5: Capture from *mobile*.

The measurement from Figure 5.5 also had a length of five minutes with an incoming call every minute. In this measurement however, *mobile* receives a text message after the second call where it receives a new TMSI. Also, *mobile* successfully establishes a connection at the end of the measurement for an incoming phone call, but crashes after. *Mobile* is assigned TMSI 0x0f22498d at the beginning of the measurement. The paging requests for the first two calls are marked red and can be found in the coloured bars marked 1 and 2. Then, *mobile* receives a text message and gets a new TMSI: 0x02224610. The GSM traffic for the text message is marked green and can be found in the coloured bar marked with a 3. At that moment, TMSI 0x0f22498d stopped appearing and paging request for TMSI 0x02224610 where detected for the 3^{th} and 4^{th} call. These paging requests can be found in the coloured bars marked 4 and 5. At the 5^{th} call, *mobile* established a connection and crashes. The paging request for the 5^{th} call followed by the GSM traffic for the call can be found in the coloured bar marked with a 6.

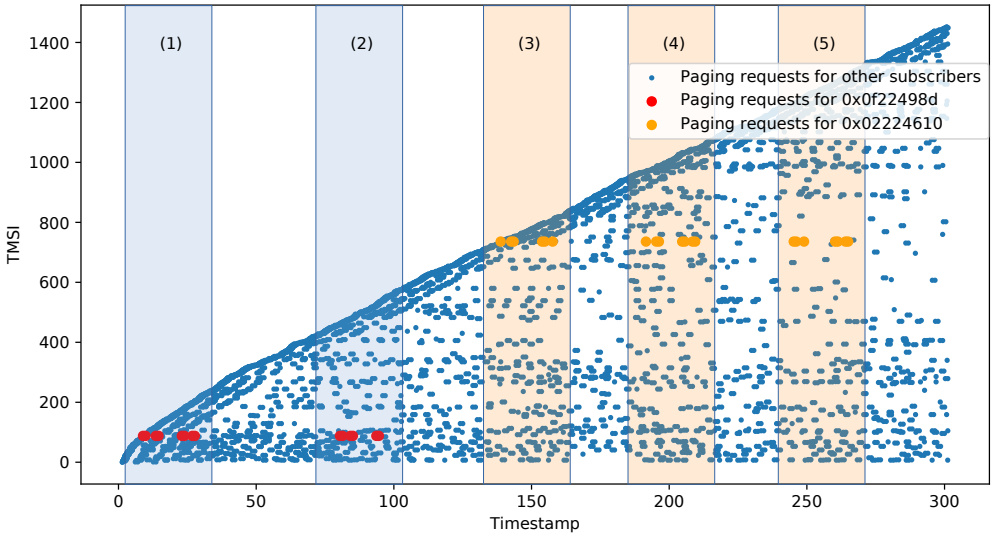


Figure 5.6: Capture from the PCH.

The measurement from Figure 5.6 was taken simultaneously with the measurement from Figure 5.5. The paging requests for TMSI 0x0f22498d are marked red and can be found in the coloured bars marked 1 and 2. The paging requests for TMSI 0x02224610 are marked orange and can be found in the coloured bars marked 3, 4, and 5. All paging requests captured on the PCH match the paging requests for the 5 calls by comparing Figure 5.5 and Figure 5.6.

Switch the phone on and and off and observe the attaching and detaching process

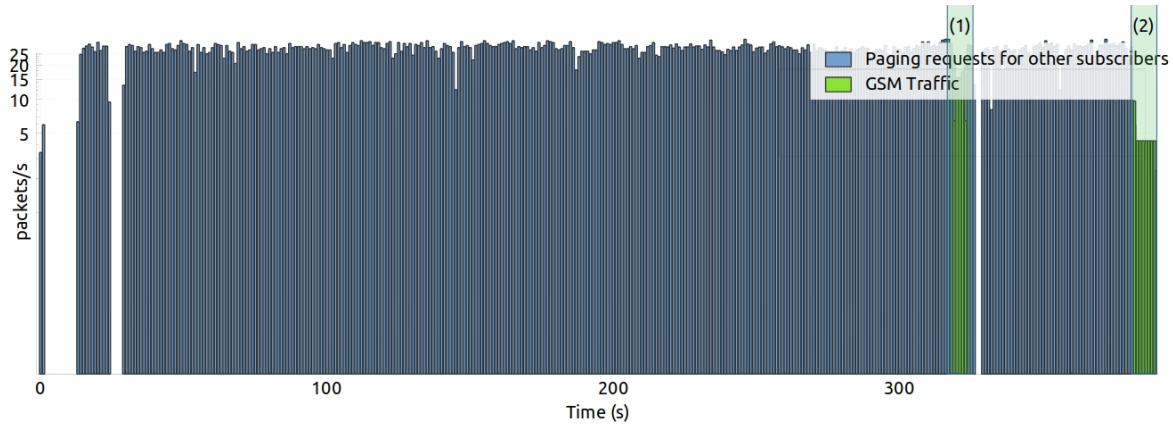


Figure 5.7: Capture from *mobile*.

During the measurement from Figure 5.7, *mobile* is switched on so that the process of attaching to the network can be observed. After around 320 seconds, *mobile* successfully attaches to the network. One minute later, *mobile* is switched off so that the process of detaching from the network is observed. The GSM traffic of the attachment and detachment process is marked green and can be found in the coloured bar marked 1 and 2, respectively. *Mobile* was assigned TMSI 0x13226501 but no paging requests for *mobile* were received.

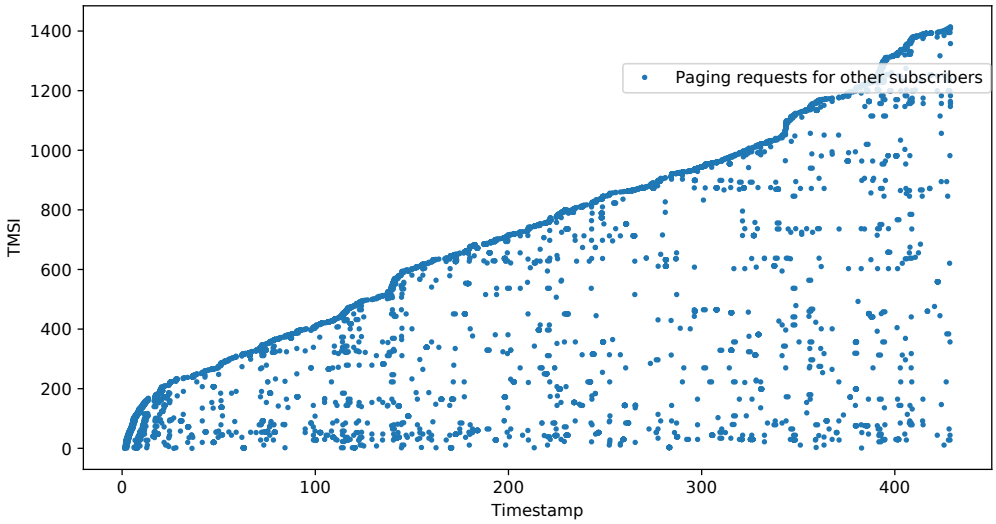


Figure 5.8: Capture of the PCH.

The measurement from Figure 5.8 was taken simultaneously with the measurement from Figure 5.7. There were no paging requests detected for TMSI 0x13226501.

Have the phone switched on and idle for five minutes

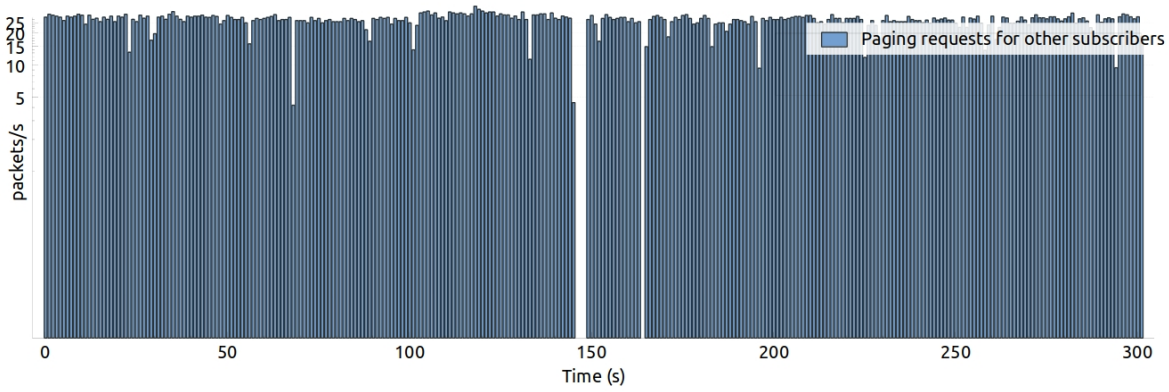


Figure 5.9: Capture from *mobile*.

The measurement from Figure 5.9 has a duration of five minutes where the phone is already attached to the network and is idle. *Mobile* was assigned TMSI 0x10224128. However, there was no traffic observed which was intended for *mobile*. There was no outgoing traffic from *mobile* either.

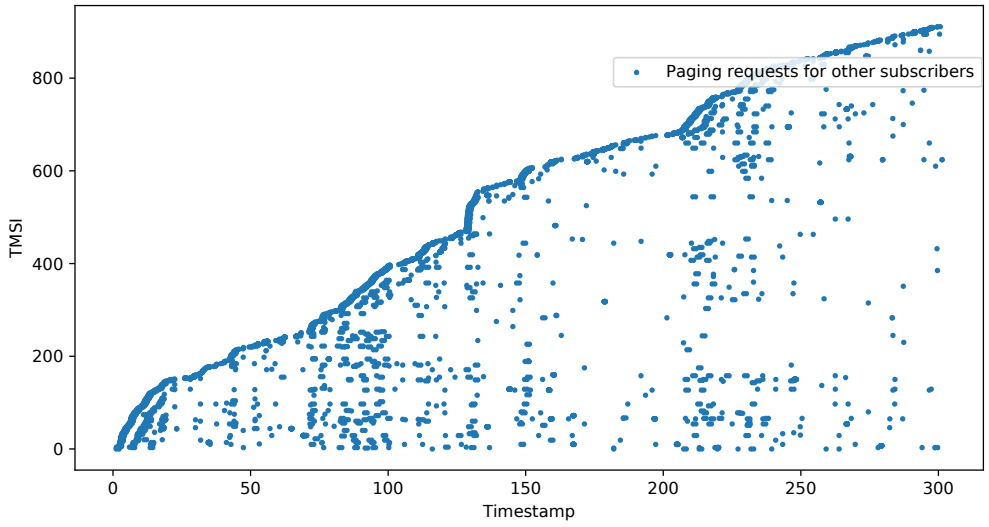


Figure 5.10: Capture from the PCH.

The measurement from Figure 5.10 was taken simultaneously with the measurement from Figure 5.9. Although not derivable from this figure, other than the absence of paging requests marked with a different colour, there were no paging requests for TMSI 0x10224128 detected.

5.4 Discussion

The previous section presents the results from measurements of three different scenarios. The first measurement has a duration of five minutes with a call every sixty seconds. The calls which are made every sixty seconds can be found in both the captures from *mobile* and the measurements from the PCH. The calls might not be right on the sixtieth second since the calls had to be made by hand. However, the measurements on the PCH match exactly with the captures of *mobile*. It seems that the paging of a subscriber happens in bursts of four paging request at the start of the call and burst of four paging requests at the end of the call. The measurement of the second scenario concerns attaching to and detaching from network. It takes *mobile* a bit longer to attach to the network, but after around 320 seconds it attaches to the network and at around 380 seconds it detaches from the network. However, the PCH is not used for the attaching nor detaching, so no paging requests for *mobile* have been detected on the PCH. Lastly, the measurement where the mobile phone is already attached to the network and is idle for five minutes is presented. There is no incoming or outgoing traffic for *mobile* captured not detected on the PCH.

Somewhat expected though peculiar, is that the mobile phone only transmits and receives signals when there is traffic from users on the network. There isn't any indication that there is any traffic between the mobile phone and the BTS regarding connection management, location updating, or something similar. This means that as long as a mobile phone is idle it is not observable through GSM. This behaviour is very unfavourable. This also means that estimating the amount of mobile phones based on the PCH would not result in an accurate number of mobile phones in a cell. Only the mobile phones with incoming traffic will be detectable. The chance that all the mobile phones in a cell are being called at the same time is very slim. Hence, PCH does not seem suitable to extract information to determine the number of mobile phones inside a cell.

With these results, two out of four research questions can answered to a satisfactory level. The research questions are: *How to detect mobile phones? How to identify a mobile phone? How to track a mobile phone? How to distinguish mobile phones?*
Mobile phones within a GSM cell can (indirectly) be detected by sniffing the *PCH*. However, mobile phones cannot be detected at all times, so not all mobile phones in the cell will be detected simultaneously. Mobile phones can be identified and distinguished by their TMSI. Each paging request contains a TMSI which is uniquely bound to one mobile phone. Because mobile phones cannot be detected at all times, mobile phones cannot be tracked. It follows that the method presented in this research is not suited to monitor crowd dynamics.

Chapter 6

Alternative Method

This research has focused on finding a new way to identify mobile phones which is suitable for monitoring crowd dynamics. Opposed to the popular probe request method using Wi-Fi, the paging requests in cellular networks were studied. Both of these methods make use of characteristics in the protocol. Wireless protocols send an identifier unencrypted at least once due to their wireless nature. This happens when a connection needs to be established, for example. A search in relevant literature yielded that crowd managing techniques have limited themselves to using existing identifiers in wireless protocols to identify and distinguish mobile phones.
This chapter proposes to look at the physical layer as an alternative method to identify mobile phones for monitoring crowd dynamics. As this chapter is solely theoretical, this chapter is intended as a strong recommendation for further research. This chapter will refer to a method presented in 2008 [4] first. Second, some more recent researches are mentioned which used a similar method in different scenarios. Finally, this chapter is summarized over the topics relevant to crowd monitoring to explain how physical layer identification can be used for crowd monitoring.

6.1 Physical Layer Identification Methods

Brik et al. [4] presented an approach which identifies, what the authors call, Network Interface Cards (NICs). This is done by looking at behaviour caused by minor variations in the circuitry of transmitters which, inevitably, arise during manufacturing. These imperfections are manifested as 'idiosyncratic artifacts' in the emitted signals. They named their method PARADIS, which stands for Passive Radiometric Device Identification System (PARADIS). This paper presents a method and a prototype. The prototype was able to identify 130 identically manufactured NICs. They solely worked with WiFi NICs, but since this method does not rely on the protocol PARADIS should work with most modulation schemes. The purpose of PARADIS is to improve authentication methods of devices in wireless networks.
PARADIS is referred to as a radiometric identification technique instead of a radio-frequency fingerprinting technique to exclude other types of RF fingerprinting techniques, like channel specific ones such as fingerprinting based on the channel impulse response.

PARADIS compares the observed signal to the theoretical ideal signal. It then composes a unique identity from the difference in magnitude of frequency, magnitude and phase errors, I/Q origin offset, and SYNC correlation of the frame. The I/Q origin offset is the distance between the origin of the ideal I/Q symbol and the measured I/Q symbol. The SYNC correlation is the correlation between the ideal and measured I/Q symbols of the signal which allows transmitter and receiver to synchronize.
The vector of these values is the radiometric signature. To map a radiometric signature to a radiometric identity, PARADIS uses a classifier. This also means that a radiometric identity is known a priori and is constructed in a training phase. The authors of this study saw their method as a security feature. In particular for countering identity forgery and abuse of cryptographic keys. PARADIS is very effective to both abuses since hardware forgery at this level is very costly and very hard to accomplish.
The authors claim PARADIS to be *simple*, *resilient* to noise because physical modulation schemes are already designed to protect data against poor channel conditions, *robust* since a radiometric identity based on hardware is hard to forge, and *ubiquitous* since it is transparent and passive, meaning that wireless devices do not communicate with PARADIS.

A more recent study used a similar method to identify wireless devices using data acquired at the physical layer [20]. St. Germain and Kragh used the Channel State Information (CSI) and a neural network as a classifier. Again, this method needs training data from the devices that it needs to identify. Also, this method is based on instantaneous CSI which means that the CSI is measured from an impulse or frequency response. As this method is designed from an authentication perspective, it might be challenging as a third party to sniff the relevant information.

Another recent study, published in 2020, successfully identified wireless devices based on their hardware components. Li et al. [21] also proved that hardware components indeed define the radio frequency fingerprint. Li et al. did this by individually replacing the local oscillator, bandpass filter, and power amplifier with identical components. Li et al. showed that the largest Euclidean distance between two fingerprints came from the reference device and the device with the local oscillator swapped. Hence, they proved again that wireless devices can be identified by the imperfections in hardware components and concluded that the local oscillator has the largest influence.
The fingerprint that Li et al. constructed is a vector of three features: time domain characteristics, the permutation entropy, and the power spectral flatness. The time characteristics are the amplitude and phase of the signal. The permutation entropy is calculated from the amplitude and phase. The authors used the permutation entropy because it can reconstruct one-dimensional time series into higher dimension vectors in multidimensional state space. The power spectral flatness is a one-dimensional value between 0 and 1 where a value close to 0 indicates concentrated power in a small number of bands and a value close to 1 indicates similar distributed power in the majority of bands.

6.2 Physical Layer Identification in Crowd Monitoring

The previous section described several methods to identify wireless devices based on unique imperfections in their hardware. Every method followed the following steps to identify devices. First, multiple features which the authors deem relevant are extracted from the signal. Examples are magnitude, phase, the error in magnitude or phase with respect to the ideal signal, I/Q origin offset, and permutation entropy. These features are normalized when necessary. Then a fingerprint is constructed in the training phase, where the fingerprint is a vector of the features. St. Germain and Kragh [20] used a impulse response to determine the CSI to use as a feature. Then, a classifier is used to identify the wireless device in the classification phase. Such a classifier could be simple, like a kNN-classifier [22] or the minimal Euclidean distance. More advanced classifiers, like neural networks, are also used.

Although physical layer identification is a different approach than the Wi-Fi probe request method and the TMSI sniffing method where existing identifiers in the protocol are used, all studies mentioned in this chapter have one characteristic in common. All methods need a training phase which requires data from the wireless devices to identify those wireless devices. The identification process is divided in two stages. The training stage, where features are extracted from a number of devices that need to be identified later. The fingerprints of these devices are constructed to create reference fingerprints. Some classifiers, like neural networks, need a rather large number of feature sets for training. In other words, the fingerprints of the wireless devices need to be known a priori. This type of training uses labeled datasets and is called supervised machine learning, which is undesirable in a crowd management scenario since the fingerprints of all users need to be determined. This requires cooperation from all users. Unsupervised machine learning allows classifiers to be trained using unlabeled datasets. A literature search in unsupervised machine learning used in physical layer identification found that this area is still unexplored. Common unsupervised machine learning methods are cluster analysis and Principle Component Analysis (PCA) [23]. Which unsupervised learning method is best for identifying wireless devices in crowd monitoring scenarios needs to be investigated in further research.

Physical layer identification has potential as an alternative way to identify wireless devices for crowd monitoring. Studies on this topic mostly aim to improve security in wireless networks. Most studies outline a scenario where the identification takes place in an authentication procedure. This is an important detail to enable supervised machine learning since the fingerprints of the devices that need to be identified is determined a priori. In order to make physical layer identification possible in crowd managing scenarios, an alternative classification method needs to be found to preserve the users not having to actively coorperate.

Chapter 7

Conclusion and Recommendations

7.1 Conclusion

This research aimed to find a different way to identify personal communicating devices, other than the common Wi-Fi probe request method. In order to find an alternative way and determine whether it is suitable for monitoring crowd dynamics, the following research questions were proposed: *How to detect mobile phones? How to identify mobile phones? How to track mobile phones? How to distinguish mobile phones?*

The GSM network has been chosen to experiment with. The GSM network was chosen because it is, to the author's knowledge, the only cellular network with an open source implementation. All other networks have publicly disclosed protocols but no open source implementations.

Using the software from Osmocom [9], GSM traffic can be sniffed. GSM uses paging requests to awake mobile phones to initiate a connection. Paging requests contain a unique identifier, the TMSI. This research conducted experiments to test if sniffing paging requests is viable for monitoring crowd dynamics. A simple DVB-T dongle was used to sniff the PCH and record all paging requests. Alongside the DVB-T dongle was a mobile phone running the OsmocomBB firmware.

How to detect mobile phones?

Mobile phones are detected by sniffing the PCH of the GSM network. This channel contains the paging requests intended for the mobile phones within the cell to awake them for incoming traffic. When a paging request was triggered for the mobile phone in the test environment by calling the mobile phone, it was successfully detected on the PCH.

How to identify mobile phones?

Mobile phones are identified by their TMSI. The TMSI is uniquely bound to a mobile phone and can be found in the paging requests. The mobile phone was detected and identified by extracting the TMSI from the mobile phone and tracing it on the PCH.

How to track mobile phones?
Several scenarios of mobile phone use were simulated. These scenarios were incoming calls, attaching to and detaching from the network, and staying idle. Note that this was not a smartphone, but a mobile phone from 2005. Paging requests were only visible at the event of an incoming call. There were no paging requests visible during attaching to or detaching from the network nor when the phone was idle. Hence, there were not enough paging requests to successfully track the mobile phone.
How to distinguish mobile phones?
Since the GSM protocol uses an identifier which is transmitted unencrypted and can be sniffed, the TMSI can be used to both identify and distinguish mobile phones.

From the experiments can be concluded that there are too few paging requests on the GSM network to consistently monitor mobile phones. Although the mobile phone could be identified and distinguished from other phones, it could not be monitored when the phone is not active due to the lack of paging requests when the mobile phone is not active. For this reason, the mobile phone could not be tracked either. This leads to the conclusion that the information on the Paging Channel (PCH) of the GSM network alone is not enough to monitor crowd dynamics.

7.2 Recommendations

7.2.1 Repeating the experiment on LTE

The method presented in this research was based on and tested with the GSM network which only facilitates calls and text messages. Modern smartphones have a lot more functionality, thus by definition a lot more network traffic. In 2020, most traffic happens on the LTE network. A strong recommendation would be to try to repeat this experiment on the LTE network. LTE also makes use of TMSIs [7] and a PCH, so it is likely that the experiment will work on Long Term Evolution (LTE) as well. The difficulty is that LTE does not have the availability of a simple setup to experiment with, like GSM does with the Osmocom project.

7.2.2 Further research on physical layer identification

This research concludes that there is not enough information on the PCH of the GSM network to monitor mobile phones. Chapter 6 looked into methods to identify wireless devices through physical properties. Physical layer identification is not bound to a protocol and works with any digital modulation scheme. Physical layer identification shows potential as an alternative method to identify wireless devices for crowd monitoring. However, all studies on physical layer identification treated in Chapter 6 use a classifier which is trained using supervised machine learning. Since this requires cooperation from the users a priori, an alternative classification method needs to be found.

Chapter 8

Discussion

This chapter addresses some remarks on this research. First, the lack of a ground truth is mentioned. Second, working with cellular networks in the research field is mentioned. Lastly, working with Osmocom is discussed.

Several studies regarding crowd management are mentioned in this research. It has become clear that there is no perfect way to monitor crowds in real time. There are studies that develop crowd managing methods [5] [2], studies that compare crowd managing methods [1], and studies that try to defeat the shortcomings in crowd managing methods [3]. All studies, including this research, suffer from the lack of a ground truth. When monitoring crowds in any way, there is no way of telling that the observations are accurate. This research sniffed the PCH in order to determine the amount of mobile phones in a cell. However, there was no way of determining the accuracy of the measurements. This is a problem that a multitude of studies suffer from.

This research worked with the GSM network because of the Osmocom project [9], a project which realised an open source GSM implementation after documentation of the Calypso chipset was leaked. This means that it did not mean to happen. To the authors knowledge, this did not happen with other cellular networks, like LTE. Only the documentation on the protocol is available, so there is no manual how to implement it. All existing chips that implement the protocol are not open source. This means that practical implementations are scarce, except the implementations by the companies that professionally manufacture the chips found in smartphones. This gives researchers which do not work for these companies a tremendous disadvantage when working with LTE, or any other cellular network but GSM. Experiences from this research revealed that cellular networks are not easy to study in contrast to Wi-Fi or Bluetooth, for example. There is a lot of consumer software and hardware to develop and experiment with these technologies. This makes a learning experience much smoother. This leads to the conclusion that it takes relatively more time and effort time to work with cellular networks if the researcher is not already familiar with the cellular network.

This research worked with tools from the Osmocom project. This project is community based, started in 2010, and has completely evolved around the rather old GSM network. Experiences from this research learn that Osmocom's software used in this research is quite fragile and that its functionality is not consistent. The GSM network is not implemented the same way in every country, which could be an influencing factor. For example, the GSM networks in The Netherlands use frequency hopping, a feature which is not used by the majority of countries where GSM is deployed. Working with a community project which works with old technology, like Osmocom works with GSM, learned that contacting the developers is strongly advised when software does not function as it should be. Developers can either help or immediately tell whether a desired feature is feasible in the set circumstances. This proved to be of great help in this research.

Bibliography

[1] A. Draghici and M. van Steen, "A survey of techniques for automatically sensing the behavior of a crowd," *ACM computing surveys*, vol. 51, no. 1, 2018.

[2] C. Chilipirea, "Crowd data analytics as seen from Wifi: a critical review," Ph.D. dissertation, University of Twente, Netherlands, 11 2019.

[3] C. Chilipirea, A. Petre, C. Dobre, and M. van Steen, "Presumably simple: Monitoring crowds using wifi," in *2016 17th International Conference on Mobile Data Management (MDM)*. United States: IEEE, 2016.

[4] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," 01 2008, pages. 116–127.

[5] N. Wijermans, C. Conrado, M. van Steen, C. Martella, and J. Li, "A landscape of crowd-management support: An integrative approach," *Safety science*, vol. 86, pages. 142–164, Jul 2016.

[6] J. D. Day and H. Zimmermann, "The osi reference model," *Proceedings of the IEEE*, vol. 71, no. 12, pages. 1334–1340, 1983.

[7] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Numbering, addressing and identification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS), 03 2020, version 16.2.0. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/23_series/23.003/23003-g20.zip

[8] "OsmocomBB Git Repository," http://git.osmocom.org/osmocom-bb/, Osmocom, 2020.

[9] "Osmocoms Project Page," http://osmocom.org/projects, Osmocom, 2020.

[10] F. van den Broek, "Catching and Understanding GSM-Signals," MSc thesis, Radboud University Nijmegen, Mar 2010.

[11] J. S. Jeffrey Bosma, "Eavesdropping on and decrypting of GSM communication using readily available low-cost hardware and free open-source software in practice," MSc thesis, University of Amsterdam, May 2012.

[12] F. L. Pönsgen, "GSM and GPRS Security Using OsmocomBB," MSc thesis, Norwegian University of Science and Technology, Jun 2015.

[13] “What is the maximum range of a Bluetooth connection?” https://www.samsung.com/levant/support/mobile-devices/what-is-the-maximum-range-of-a-bluetooth-connection/, Samsung, August 25, 2020.

[14] “Official industry association Website,” https://www.wi-fi.org/, Wi-Fi Alliance, 2020.

[15] L. Schauer, M. Werner, and P. Marcus, “Estimating crowd densities and pedestrian flows using wi-fi and bluetooth,” *MobiQuitous 2014 - 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 01 2014.

[16] “Map of base station locations,” https://antennekaart.nl/kaart/2g?lat=52.21932116&lng=6.89327717&zoom=16&providers=kpn,t-mobile,vodafone, Antennekaart, August 25, 2020.

[17] “RealTek RTL2832U Product Page,” https://www.realtek.com/en/products/communications-network-ics/item/rtl2832u, RealTek, 2020.

[18] “gr-gsm Git Repository,” http://git.osmocom.org/gr-gsm/, Osmocom, 2020.

[19] “Wireshark Version 3.2.3,” https://www.wireshark.org/, Wireshark, August 25, 2020.

[20] K. St. Germain and F. Kragh, “Physical-Layer Authentication Using Channel State Information and Machine Learning,” 06 2020.

[21] Y. Li, X. Chen, Y. Lin, G. Srivastava, and S. Liu, “Wireless Transmitter Identification Based on Device Imperfections,” *IEEE Access*, vol. 8, pages. 59 305–59 314, 2020.

[22] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pages. 21–27, 1967.

[23] I. Jolliffe, “Principal component analysis. 2nd ed,” *http://lst-iiep.iiep-unesco.org/cgi-bin/wwwi32.exe/[in=epidoc1.in]/?t2000=017716/(100)*, vol. 98, 10 2005.