Doctoral Dissertation

Doctoral Program in Electrical, Electronics and Communications Engineering

(34^{th}cycle)

MEC-based Mobility Tracking and Safety Service through IoT

By

Kalkidan Gebru

******

Supervisor(s):

Prof. Carla Fabiana Chiasserini, Supervisor

Prof. Claudio Ettore Casetti, Co-Supervisor

Prof. Paolo Giaccone, Co-Supervisor

Doctoral Examination Committee:

Prof. Antonella Molinaro, Università Mediterranea di Reggio Calabria

Dott. Alessandro Nordio, IEIIT-CNR, Torino

Politecnico di Torino

2022

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Kalkidan Gebru
2022

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

Let this be for the people God gave me.

Acknowledgements

First and foremost, I would like to acknowledge and thank the eternal novelist, God the Father; the professor of life, God the Son; and the publisher of miracles, the Holy Spirit. He gave it all and keeps giving me more. I thank God for he has favoured me with people who care for me, including my advisors on this work.

I would like to acknowledge and give my sincere gratitude to my advisors, Prof. Carla Fabiana Chiasserini and Prof. Claudio Ettore Casetti, for they have accepted me since my masters and have never stopped their support. I would like to thank them for their patience and guidance. They have encouraged me through tough times, paving the way for me to complete the PhD program. It is a pleasure for me to know them, let alone collaborate with them. I would also like to thank Prof. Paolo Giaccone for being always there when I needed him, dedicated to guiding and supporting me. I am very grateful for the time he spent enlightening me.

My heartfelt gratitude goes also to my parents, grandparents, and spouse. Their prayers and blessings gave me the courage to go forth in life.

I would also like to thank the members of Luce di Cristo in Torino. They are true friends and family to me, and will remain in my heart forever. They have made my days in Torino delightful and unforgettable. They are my blessings.

I would also like to thank Giuseppe, Marco, Greta, and Corrado for their support and cooperation.

Abstract

Monitoring people afoot as well as vehicles has become crucial, not only for safety but also for several practical business applications, facility management, and services. In relation to that, the proliferation of IoT-based services and the growing industry of telecommunications are playing a vital role in providing the perfect ecosystem for advanced smart city use cases. Numerous research and studies supported by the private sector are addressing the various use cases for mobility tracking and safety services. The purpose of this work is to make contributions to the mobility tracking and safety services of smart cities with the help of IoT devices and telecommunication infrastructures. Hence, IoT based WiFi sensors and MEC based virtual sensors were used for mobility tracking and safety systems in our work.

The WiFi sensor devices detect the presence of people from the WiFi signals, the WiFi probe request frames of smartphones. We have considered two types of devices, namely, commercial, off-the-shelf WiFi scanners and ad-hoc designed WiFi scanners implemented with Raspberry PIs. They provide different levels of visibility of the captured traffic. The detected probe request packets contain the associated MAC address of the transmitting device. Since the MAC address is considered personal data by the EU GDPR, a privacy protection mechanism was required for tracking people's movements. Although the currently available technologies have made efforts through anonymization techniques, the privacy concern remains vulnerable for MAC addresses. Thus, we have implemented a privacy-preserving scheme for addressing the privacy challenge and tackled the problem of identifying people's movement for the popular mobility patterns in an urban environment by using WiFi sensors connected to the cellular network. Furthermore, events and group activities were captured with the support of a model. We illustrate our approach and present results derived from live measurements in a testbed deployed in the city of Turin

within the 5G-EVE project.

On the other hand, we have implemented a MEC-based EVS safety service, in particular collision detection on intersections for vehicles and pedestrians. The system leverages the mobile network, collecting mobility data (i.e., position, velocity, acceleration, etc.) periodically from smartphones and onboard units of vehicles in order to have awareness about the monitored area. The EVS system is capable of detecting collisions ahead by computing the future trajectories of all the vehicles and pedestrians in a given geographical area. When imminent collisions are detected, warning messages are sent to the regarding entities before the impact, so collisions can be avoided. Furthermore, in the work, traffic flow scenarios were modeled for an urban environment. The testbed of the system was carried out on the OAI standard platform. Finally, we present the evaluated performance of the mobility safety system.

Contents

List of Figures

Chapter 1

Introduction

Nowadays, smart cities are becoming more common in most parts of the world. The reason behind that is the availability of Internet-of-Things (IoT) technology and its impact on the performance of services. One of the pillars of a smart city is the mobility system [3]. For this reason, the growing sector of telecommunications and IoT services have combined to provide the perfect platform for advanced smart city use cases. IoT devices can gather and share data about the environment, while the 5G technology facilitates quick data delivery and response services to support mobility-based services.

To improve the lives of people through efficient service delivery, it is first important to understand their needs and demands. One of these needs is systematic mobility support to accomplish their daily activities. To achieve this, however, detailed mobility surveys are required so as to understand the behaviors of the environment. It is essential to have information on people's mobility, such as where they would like to go and which route they would like to take frequently through analysis. Thus, tracking their movements as a means to getting such information is quite beneficial.

Intending to track mobility, it is crucial to have an infrastructure suited for detecting events in the environment. This infrastructure should have low cost and energy consumption as much as possible, but most importantly, it must sustain people's privacy. Therefore, we will track and monitor mobility through IoT technologies and mobile network support. IoT devices collect mobility data from their environments

and forward the data to a secure server through cellular networks. Then, data can be processed and analyzed from a remote station to study people's daily activities.

1.1 Mobility monitoring and tracking system

In this work, we are focused on tracking the movements of people in an urban environment to improve service quality. With that in mind, we can observe the main practices that should be carried out in tracking mobility. The mobility monitoring process in a smart city mainly includes three related phases in general, as shown in Fig 1.1.

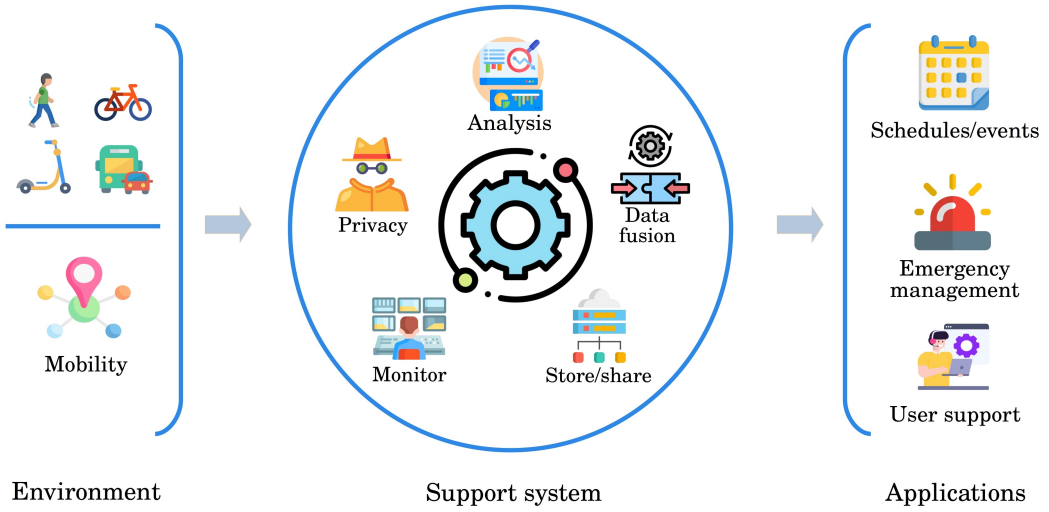


Fig. 1.1 Mobility support process

In the first phase, proper information is gathered with the help of IoT sensors. Such information can be collected from the signals of smartphones, which could be hand-held or inside a pocket during mobility. For this work, we will consider these signals, in particular the WiFi probe request packets that are emitted with or without users' awareness. Details of these signals are discussed in Sec 1.2.

After gathering data through IoT sensors, the second phase deals with management and support services. In this phase, the privacy of people must be protected primarily.

Thus, any information that links to their identity must be secured through privacy-preserving techniques. Afterward, the mobility monitoring process can trail intense mobility analysis. Although not included in the current work, further data fusion processes can take place, i.e., integration of other environmental data like air/weather conditions can be tied with mobility. While these operations are executed, the data should be stored properly. When needed, the data can be shared between the rightful owners.

Following the completion of the most significant operations, there could be third-party applications to support both service providers and service users. These applications can provide helpful information to the service providers and users with the help of the second phase procedures. From the application services, schedules and events can be reported, emergencies can be notified, and any mobility-based service can be dispatched for users.

1.2 WiFi probes for mobility tracking

The mobility tracking methodologies proposed in this thesis are focused on the WiFi probe request packets, thus understanding how the protocol works is fundamental.

Before having a WiFi traffic service, a connection procedure must succeed between a WiFi access point and client devices, such as smartphones and laptops, according to the 802.11 WiFi protocol. Among various types of management, control, and data frames used in the protocol, the WiFi probe request packets are part of the management frame exploited for tracking mobility. For any device to join a WiFi network, it needs to carry out the authentication and association process by exchanging management frames as defined by the protocol. The procedure, also referred to as scanning, can be based on either active scanning or passive scanning of nearby networks.

In the course of active scanning, the client device is responsible for initiating the network identification process, before the authentication and association, by transmitting probe request packets and listening for a probe response packet from the

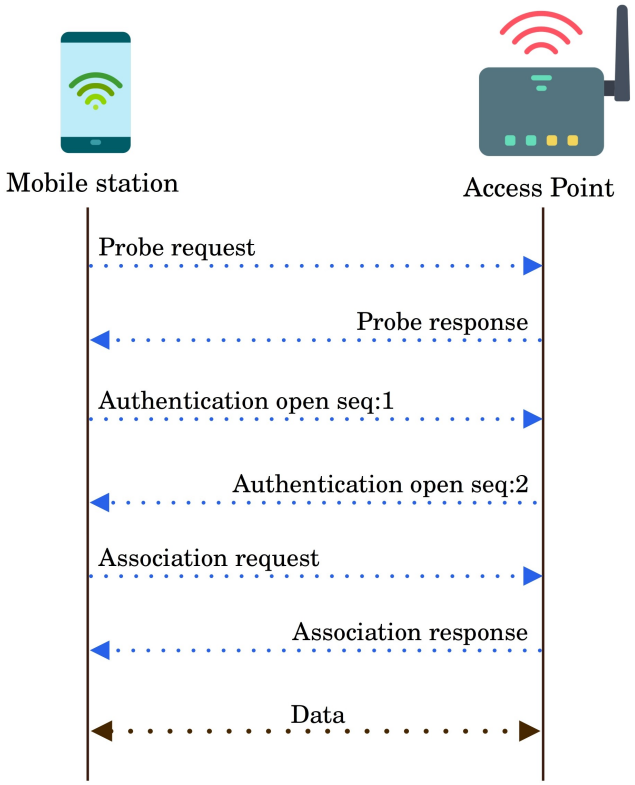


Fig. 1.2 Management frame exchange between a client device and an access point

nearby access points. The probe requests contain information such as the source and destination addresses, the SSID of the known network, and the supported data rates of the client device, mainly. While the source address is the MAC address of the device, the destination address is set to 'ff:ff:ff:ff:ff:ff', thus the packet is broadcast. After receiving the probe packets, the access points in range can notify the client device the possibility of joining the network through the probe response packet, if at least one of the advertised data rates is supported by the network. Hence, the authentication and association procedures can take place through authentication request/response frames and association request/response frames, respectively. Fig. 1.2 shows the complete process of the identification, authentication, and association process before the data exchange between a client device and an access point.

During passive scanning, the client device does not send any probe packets to identify the network. Instead, the station remains in listening mode for a specific amount of

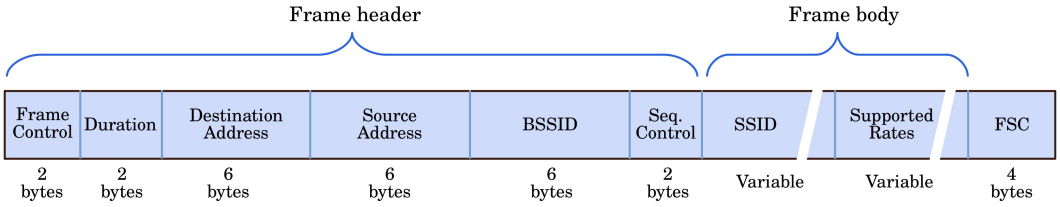


Fig. 1.3 WiFi probe request frame [1]

time, waiting for beacon frames broadcast by access points on one of the supported channels. After a certain time, the client device will switch to the next supported channel, repeating the process until a connection is established with an access point.

Among the two network identification techniques, the active scanning method is preferred by client devices most commonly due to the accelerated network association and lower energy consumption benefits. Furthermore, hidden networks can disclose presence information only to the stations sending probe request frames. These frames should also incorporate the SSID of the hidden network.

As reported by [4], the probe request frames, to be used for mobility tracking, have a transmission rate ranging from 55 to 2000 per hour depending on the vendor and the status (e.g., sleep mode) of the mobile devices. The transmission rate does not have any defined standard, so vendors and applications managing the device are in charge of how frequently the probes should be generated. When we consider a tracking system, mobility can be detected from these frames, but it is not possible to have confidence in the transmission rate of the frames. Although extracting valuable information from the packets raises privacy concerns, smartphone OS vendors such as Android and iOS have put into practice a tracking countermeasure, a MAC address randomization technique. After the execution of the privacy protection mechanism, when the probe frames are broadcast from the client devices, the original MAC address is temporarily replaced with a randomly generated MAC address. Therefore, tracking mobility requires the best efforts in order to formulate significant information about people's movement through different techniques.

1.3 Related works

When we consider people's mobility monitoring systems, several studies with different goals, have proposed various solutions using WiFi probes and other techniques. Among these works, [5] focuses on building knowledge from WiFi probes targeting major areas where classification can be applied. [6] targets crowds attending exhibitions. By mounting WiFi/Bluetooth sensors equipped with directional antennas on the ceiling, localization and crowd density estimation were addressed with video ground truth support. Before processing, a clustering technique based on the duration of exhibition hours was also used to identify mobile visitors from stationary devices. The work on [7] uncovers social relationships based on semantic trajectories, which are spatial and temporal-based trajectory patterns. The authors have proposed a classification algorithm to extract the resident population of a building. A cluster is decided if users are following peak time patterns on the scanner near the building door. In [8], the use of different buildings and movement patterns between them is monitored within campus. Groups within a building are identified based on their length of stay. The relationship among the buildings was based on mobility patterns. Ground truth data was collected from manual counting of people using the buildings. [9] shows the possibility of distinguishing geographical origins from SSIDs of collected WiFi probes. The authors' methodology was based on linking Preferred Network Lists (PNL) with geographical coordinates from the Wigle database. From the snapshot of crowds, they were able to predict official nationwide voting results. Different scenario is analyzed as well using GPS data to classify movements and stops based on threshold parameters for maximal movement and minimal stop durations [10].

The author of [11] constructed solar-powered WiFi scanners around a large infrastructure-free camp; as a result, they performed movement analysis between locations near the scanner and were able to count arrivals and departures for the specific event. However, the problem of randomization, more on infrastructureless camps, and lack of ground truth data makes it incomplete. Instead [12], besides just using mobile phones, it presents a way of retrieving the behavior of groups in a crowd using dedicated WiFi badges that emit probes systematically. This was suggested to address the uncontrolled probe transmission behavior of phones, which is vendor-based. Alternatively, Antonio et al. present a mobility monitoring system based on data collected

from Bluetooth for urban environment scenarios [13]. The authors' aim is to detect jams, trace routes, and compute the average speed of vehicles. In addition, [14] focuses on estimating traffic volume between points and over multi-points while maintaining the privacy of vehicles in the traffic data record. The estimation relies on the number of encoded vehicles at each point and the number of intersections between the traffic records over time.

The study on [15] presents a technique to estimate the number of footfall by clustering probes based on time and sequence thresholds, which are the maximum time difference and maximum sequence number difference between probes, respectively. [16] presents mobility monitoring on a customized architecture with stations and a cloud. While the stations are used for collecting and partial data processing, the cloud (Google cloud platform) is used for virtualization and data management and processing. Using the architecture, the authors aim to address mobility monitoring by focusing on return time, permanence time, and density on a site. The paper [17] intends to use WiFi technology for indoor localization. The RSSI measured by multiple sensors is used for training and location estimation. The authors have also performed data fusion between the motion of the smartphone and a map of the indoor place. [18] examines the technology for estimating number of people during flows. The survey is collected to determine the percentile of people who enabled WiFi, and this percentile was used to convert WiFi counts into estimated number of people. In order to cover the target area more accurately, [19] presents grid monitoring methodology for a hub. The methodology used the Kalman filter for estimating RSSI in a position and fitting curve equation to estimate the relationship between RSSI and distance of the signal.

Several other works in the literature, primarily in the automotive sector, have also considered mobility safety services, such as [20]. For instance, [21, 22], suggested collision detection and avoidance systems without the support of any cellular infrastructure. The work presented in [21] aims at collisions between vehicles and pedestrians in industrial plants, with no specification of what type of wireless communication technology is used. Instead, [22] aims to improve the response time of first aid by using a smartphone accelerometer in order to automatically detect collisions after the incidents. The work in [23] proposed a collision detection application based on cellular infrastructure vehicle-to-vehicle communications. The authors

consider only continuously active direct communication between vehicles, which is not always available in urban environments.

In recent times, the mobile network has become the main supporting infrastructure for the automotive sector. Various works, such as [24–26], have compared the 4G-LTE network to the 802.11p network, and [26] concludes that the mobile network is superior to the 802.11p network. Instead, [25] considers the cellular network to be inconvenient for a collision detection system because of the handoff procedures and the Doppler effect. Nevertheless, the controversy remains in the research world.

1.4 Structure of the thesis

In this work, we have used two types of sensor systems for monitoring mobility in a given geographical area. The first one is an IoT-based WiFi sensor, while the second one is a virtual sensor system. The next three chapters (Chapter 2, Chapter 3, and Chapter 4) focus on WiFi-based sensor systems, while the last chapter (Chapter 5) covers the virtual sensor system based on mobile networks.

The organization of the following chapters includes a privacy-preserving technique for WiFi-based mobility tracking systems, Chapter 2, where off-the-shelf and ad-hoc designed WiFi sensors are used for detecting the presence of individuals. The sensors can collect WiFi probe request frames from mobile devices, in which the frames include the MAC address of the device. Since the MAC address is considered personal data by the EU General Data Protection Rule (GDPR), a privacy-preserving mechanism is required to collect and process the WiFi probes used in the tracking system. After addressing the privacy challenges in Chapter 2, the next chapter, Chapter 3, focuses on the mobility tracking system. In this chapter, we emphasize mobility tracking methodologies. Along with that, we use the 5G-EVE architecture for the mobility testbed. Furthermore, ground-truth experiments are made for mobility flow scenarios. In chapter 4, statistical analyses are made based on the collected WiFi probes and the associated MAC addresses. In addition to that, a model is used to detect groups and related events in the environment. The final main chapter of the thesis, Chapter 5, unlike the previous chapter, is based on a virtual sensor system

implemented as a MEC service. The work focuses on delivering a mobility safety service for vehicles and pedestrians by detecting collisions ahead.

Chapter 2

A Privacy-preserving Scheme for Passive Monitoring of People's Mobility through WiFi Beacons

Part of the work presented in this chapter has been published in [27, 28]:

- *Kalkidan Gebru, Marco Rapelli, Riccardo Rusca, Claudio Casetti, Carla Fabiana Chiasserini, Paolo Giaccone, "Edge-based passive crowd monitoring through WiFi Beacons," Computer Communications, Volume 192, 2022, Pages 163-170,ISSN 0140-3664, https://doi.org/10.1016/j.comcom.2022.06.003.*
- *K. Gebru, "A Privacy-preserving Scheme for Passive Monitoring of People's Flows through WiFi Beacons," 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), 2022, pp. 421-424, doi: 10.1109/CCNC49033.2022.9700591.*

Analyzing people's movements in urban environments is central to several critical applications related to safety, as well as to a plethora of convenience services designed for mobile users (e.g., car sharing, use of public transports, and store recommendation systems). In particular, for many applications, it is essential to detect the pattern taken by people's flows at different times of the day/week. One of the key technologies to achieve this goal is the IoT [29, 16], as IoT devices are becoming pervasive and most of them are equipped with a radio interface, such as WiFi, that can conveniently connect them with other devices as well as with

the communication network infrastructure. Furthermore, they typically consume little energy, hence they contribute to creating sustainable communication systems, have low cost, and pose less privacy issues than other devices like smart city cameras.

In this part of the work, we are interested in preserving users' privacy during data collection and processing. In particular, we focus on exploiting both commercial sensors and such simple devices as Raspberry PIs, equipped with a WiFi interface. Such devices can scan the WiFi spectrum for probe requests, i.e., packets transmitted by user hand-handled devices towards nearby access points. Using the logs provided by these spectrum scanners, we develop techniques to ensure that the collection and processing of the data meets the GDPR [30].

2.1 Data collection for WiFi based passive monitoring

To monitor people's mobility, data can be collected from different sources, such as cameras. However, the focus of this work is on collecting the WiFi signals that are generated by mobile devices so that meaningful mobility information can be developed. Since individuals' privacy must be protected primarily, both off-the-shelf and custom-built WiFi sensors filter signals, collecting only the WiFi probe request packets required for tracking.

To study and support mobility in a smart city, two off-the-shelf devices are installed at the Politecnico di Torino campus in Torino, Italy, by the collaboration of *Telecom Italia* (TIM) and *Comune di Torino* for the 5G deployment project. These devices, henceforth referred to as sensors and scanners, are capable of sensing the WiFi signals that are transmitted from nearby mobile devices. One of the scanners, labeled X on Fig. 2.1, is located on the street of *Corso Castelfidardo*, at the last gate of *Politecnico di Torino* when facing the *Porta Susa* train station, thus covering the activity close to the gate mostly. The second scanner, labeled Y on Fig. 2.1, is placed on the street of *Corso Castelfidardo*, attached nearby another gate of *Politecnico di Torino*. This gate is the first next-door to the *Liceo Scientifico Galileo Ferraris* high school. The two scanners can process only the WiFi probe request packets that are broadcast through the active scanning process, so they will only listen passively. On top of filtering the WiFi probe request frames, these sensors also hide parts of

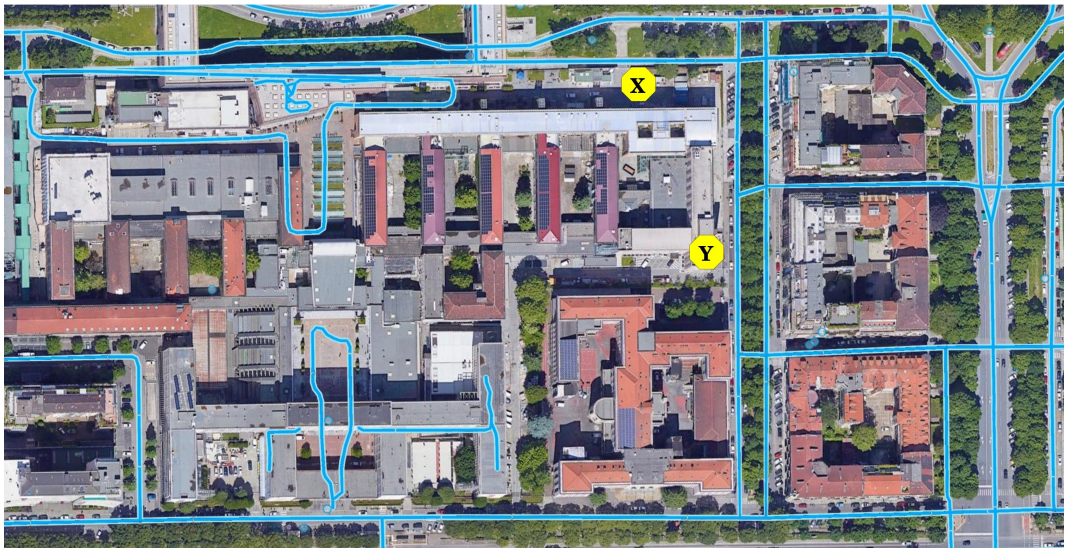


Fig. 2.1 Mobility environment and scanners

the frame, presenting only the MAC address of the device (hashed) and the vendor information extracted from the first three bytes of the MAC address. This means these off-the-shelf scanners are purposed for counting devices nearby since the only considered part of the frame is the MAC address. Other than the hashed MAC address and the vendor information, the received signal strength (RSSI) and the timestamp are recorded per frame upon reception. Fig. 2.2 shows this information in JSON format.


```
"data":[{"RSSI":"-68","Vendor":"Unknown","TimeStamp":"2019-04-15
12:32:45","MAC":"B7800478990D1DB2BF3E60B491F0FA62DD4AF1D199382331A01EE5BA"},
{"RSSI":"-64","Vendor":"Unknown","TimeStamp":"2019-04-15
12:32:45","MAC":"9EBA73946A39D741DAC93C6AAB80182FD567088541B3ACCC12332901"},
{"RSSI":"-1","Vendor":"Unknown","TimeStamp":"2019-04-15
12:32:45","MAC":"3F21F524CDE802F7621A3B8223F932162F319808830AC08F144314FA"},
{"RSSI":"-1","Vendor":"Unknown","TimeStamp":"2019-04-15
12:32:45","MAC":"17285C38E98866838A2B28801BAAF907E6BAAA3BA8629078BC8AFCDD"},
{"RSSI":"-1","Vendor":"Unknown","TimeStamp":"2019-04-15
12:32:45","MAC":"B57A06FC0CDC4291DD237AC269FD0CEA5E60D4B9B5ACD149EC5FD759"},
{"RSSI":"-86","Vendor":"Apple","TimeStamp":"2019-04-15
12:32:45","MAC":"9E1732ACFBA05D4CE26E77F74370E30F568884EA09412255383A0BFB"},
{"RSSI":"-1","Vendor":"Apple","TimeStamp":"2019-04-15
12:32:45","MAC":"ACBA1DF88216CC19B3A07A5B4B5C75A8AB7779E31871A22B551FCC54"}]
```


Fig. 2.2 Log sample of probe requests recorded by the off-the-shelf WiFi scanners

The second ad-hoc designed WiFi sensors, instead, are aimed at collecting the WiFi probe request packets along with all the fields of the frame. Hence, data processing is more practicable on these types of sensors than on commercial ones. Note

that, the purpose of collecting such information is to support the mobility system, thus people's privacy must be taken into account. For that reason, a necessary privacy-preserving scheme has been implemented to deal with such types of sensors. Sec. 2.3.2 explains further how data is managed on such types of sensors.

2.2 Privacy challenges

The first and most important part of WiFi-based mobility monitoring is the data collection. These days, however, data collection is not allowed completely due to privacy reasons, and for those allowed, there are privacy protections made by the law as well as the technology developers. For that reason, there are two main privacy challenges when considering data collection and processing on WiFi-based monitoring.

2.2.1 General data protection

In the first place, concerning the law, there are regulations, such as the GDPR set by the EU, that are challenging for data collection and processing. These regulations apply to any organization or body intending to control or process the collected data. The goal of such regulations is to give people control over their personal data. Hence, the GDPR has defined personal data as "any information relating to an identified or identifiable natural person (data subject)". Today, most communications and services require personal information that could reversely identify individuals. When we consider the WiFi-based mobility tracking, which is based on the probe request packets, the most sensitive information is the MAC address, while the remaining fields do not pose any threat to users. Although the other fields, such as the sequence numbers and tagged parameters extracted from the Information Element (IE) of the probe packets, could be leveraged [15], they ultimately rely on the MAC address for the association. Therefore, since the primarily important field of the WiFi probe request frame for tracking, the MAC address, is considered personal data by the GDPR, the data process and management cannot be at will.

On the other hand, processing mechanisms such as pseudonymization and encryption, where the personal information is replaced by another unique identifier, could be

considered. However, these techniques remain vulnerable since they can be used to re-identify a person, so MAC remains personal data. The other technique pointed out by the GDPR is anonymization, with the idea of anonymized data being irreversible. Thus, data such as MAC addresses are no longer considered personal data. To support anonymization, there are modern hash functions that digest personal data. Such mechanisms are implemented by off-the-shelf scanners, for instance, the Meshlium WiFi scanner. However, whether these techniques are sufficient on their own or have met the GDPR requirement remains an open question. Therefore, further explanation is given in Sec. 2.3.1 when we look at the anonymization technique implemented by the off-the-shelf scanners.

2.2.2 Device's privacy

Apart from the regulations, there is another challenge implemented by mobile devices, where personal data is completely hidden as a privacy protection mechanism. This is done when sending probe request frames. Instead of the original personal data (MAC address), a randomly generated MAC address is used. As a result, the technique is known as randomization. When devices employ randomized MAC addresses in addition to the original, there is going to be bias in the WiFi-based mobility monitoring system. In other words, while a device is supposed to be counted once, it is reported more than once with different identifiers (MAC addresses). Therefore, such personal data protection methods have extreme challenges for the WiFi-based mobility monitoring system.

2.3 Privacy-preserving schemes of WiFi sensors

As stated in Sec. 2.1, we have leveraged both off-the-shelf Meshlium WiFi scanners and ad-hoc designed WiFi sensors. The first device has its own privacy-preserving scheme implemented by construction, specifically the anonymization technique. However, the privacy-preserving scheme implementation of the second device, the ad-hoc designed WiFi sensor, is up to the designers since access to the data is unrestricted on such devices. The privacy-preserving scheme of the two WiFi sensors is explained in the next two sections.

Meshlium Iot gateway
commercial scanner

Raspberry Pi 3 Model B for
ad-hoc scanner

Fig. 2.3 Off-the-shelf and offhand devices used for the WiFi sensors

2.3.1 Anonymization in commercial WiFi sensors

The WiFi probe request frames from nearby devices are collected by off-the-shelf WiFi sensors. However, data access is strongly restricted. The only information recorded from the received frames is the MAC address in digested format. Although there are three additional fields attached to it, they are not part of the WiFi probe request packets. These three fields are (*i*) the vendor information, (*ii*) the RSSI, and (*iii*) the timestamp, as seen in Fig. 2.2. While the received signal strength and the timestamp are measured and tagged with the reception time by the scanner, respectively, the vendor information is extracted from the first three bytes of the MAC address before the digesting process. In order to reveal the name of the vendor, the first three bytes of the predigested MAC address are searched on the local database of the scanner. When doing this, if the vendor is not on the local data list of the known vendors, the vendor information is labeled as *Unknown*.

As observed so far, one step of the privacy-preserving scheme of the scanner is to drop most of the fields of the probe request frame when storing. The other footstep of the scheme, which could be more important than the previous, is the digesting of the MAC address since it is considered personal data by the EU GDPR. In order to obscure this personal data irreversibly, the scanner has leveraged the SHA-224, thus anonymizing the detected source MAC address of the transmitting device. Since

anonymized data is no longer considered personal data according to the GDPR, the procedure has met the required criteria. However, that is not the case with the MAC address. It is true for hash functions to be irreversible in several scenarios. Yet, when it comes to digesting MAC addresses, even state-of-the-art hash functions cannot guarantee privacy as expected and considered by the EU GDPR as well.

The weak point of the anonymization-based privacy-preserving technique lies in the MAC address itself. Since the MAC address is 6-bytes in memory, the maximum possible number of addresses that can be used is 2^{48}. This number can even be less because the first three bytes represent vendors, which were about 32,207 in number until this year, 2022. This means the total number of MAC addresses in the world is no more than 2^{39}, i.e., 2^{15} for vendors and 2^{24} for device identifiers. Hence, the search space of the MAC address is 2^{39}, which is limited, if not small, for today's computing machines. With this bounded list, someone dedicated can reverse the anonymous identifier back to the original MAC address. Therefore, in order to preserve people's privacy when dealing with MAC addresses, a better approach should be considered when designing. For this reason, Sec. 2.4 explains an enhanced solution for the privacy challenge of the MAC address.

2.3.2 Privacy in ad-hoc designed WiFi sensors

For the second type of WiFi sensors, Raspberry PI devices are exploited as ad-hoc designed sensors to detect people's near-by presence. These off-hand sensors are configured to resemble the first off-the-shelf scanners, so a USB dongle antenna is used for capturing the WiFi probe requests. The RP devices are of model B, having a 1.2 GHz 64-bit quad-core ARM Cortex-A53 CPU and 1 GB of RAM, along with the Linux operating system. Although they support 802.11n WiFi, the onboard antenna does not recognize a monitor mode to capture the WiFi management packets, thus the necessity for the USB dongle antenna.

One of the benefits of having these devices is their cost, in which the entire amount was no more than fifty euros. The other most important advantage of these sensors over the commercial WiFi scanners is the unrestricted data access of probe request frames, in which more fields, such as the sequence number and tagged parameters,

can be used in addition to the original MAC address. This supports addressing one of the privacy challenges, the randomization problem. Therefore, by using this data, a de-randomization procedure is implemented while maintaining the individuals' privacy.

It is quite common now for devices with WiFi cards to use the randomization technique as a countermeasure for privacy breaches to avoid being tracked. This is a challenge for a mobility monitoring system since the devices use more than one MAC address during the scanning phase. Having more than one address per device causes counting bias during the mobility monitor and tracking. To tackle the problem, it is quite useful to have access to the received MAC address before anonymization. Note that anonymization is mandatory as set by the GDPR to protect privacy. For instance, off-the-shelf WiFi scanners such as Meshlium extract the vendor information from the first three bytes of the MAC address before anonymization.

The de-randomization procedure is based on local processing of the MAC addresses prior to the mandated anonymization. The Raspberry PI devices were handy for designing ad-hoc WiFi scanners that capture detailed information about the received probe request packets. However, this is a violation of the user's privacy since the MAC addresses are considered to be personal information by the GDPR. At the same time, it will be impossible to separate the randomized MAC addresses from the original ones unless the received probe packets are processed before the anonymization, like the previously mentioned off-the-shelf scanner Meshlium.

For the de-randomization scheme, the proposed methodology in [31], which targets counting people on a public transportation system, is considered. The scheme mainly exploits the temporal correlation of the data which is included in the header of the probe packet, such as the sequence number, since it is incremental and cyclic to be matched with a likely device [31, 15].

In the de-randomization process, firstly, WiFi traffic is captured by the ad-hoc scanner, but only the probe packets that can be identified with their sub-type tag parameter of 0×04 are temporarily stored on the local buffer with a timestamp. The header data is then processed to generate a classifying probabilistic score. The score is

based on the incremental and cyclic sequence numbers of two probe packets; thus, if the difference in the sequence numbers is acceptable in the limited time frame, the additional tagged parameters in the IE field are checked for a match. After that, the probabilistic score is extracted from the difference between the sequence numbers and the time difference between them. Hence, the higher the score, the more likely it is that the two MAC addresses belong to the same device. In accordance with the GDPR, the MAC address is anonymized after de-randomization using the SHA-224 hash function.

2.4 Secure anonymization for MAC addresses

From Sec. 2.3.1, we have seen the limitation of the anonymization of the MAC addresses, where the main problem resides in the MAC address itself. Since the search space of the addresses is limited, an anonymized MAC address can be mapped to the original address. Thus, the approach requires further improvement to preserve people's privacy during mobility.

Since the MAC addresses, including the randomized ones, are fully controlled by the vendors during the transmissions, it is impossible to have more MAC addresses than those that are out there. Therefore, after receiving the source address from the devices and before directly applying the hash function, the source address can be modified in a way to increase the search space. Keep in mind that we are interested in monitoring mobility but at the same time addressing privacy concerns. Hence, during the source address modification, the procedure must be common for all the MAC addresses to sustain the mobility monitoring. In addition, the storage space for the information must not grow any larger.

When we consider SHA-224 for anonymization, for instance, regardless of the bit size of the MAC address, it will consume 28-bytes of memory for each. The mentioned hash function, by construction, has a much larger search space when compared with the MAC address. The proposed methodology is based on this difference in the search spaces between the MAC addresses and the hash functions. While the maximum search space for the MAC address is 2^{48}, it is 2^{224} for the hash function, leaving 2^{176} slots idle permanently. The strategy is to exploit these available idle

slots. According to our example, the idle slot will support up to 22 additional bytes. The ideal solution is to compensate the idle slots with a security string, referred to as a password, of length at least 22-bytes for the SHA-224. Therefore, whenever a probe packet is detected, the MAC address is chained with a password before the digest procedure. The chaining process may include shuffling the MAC address with the password in any format, or just concatenating the two in any order.

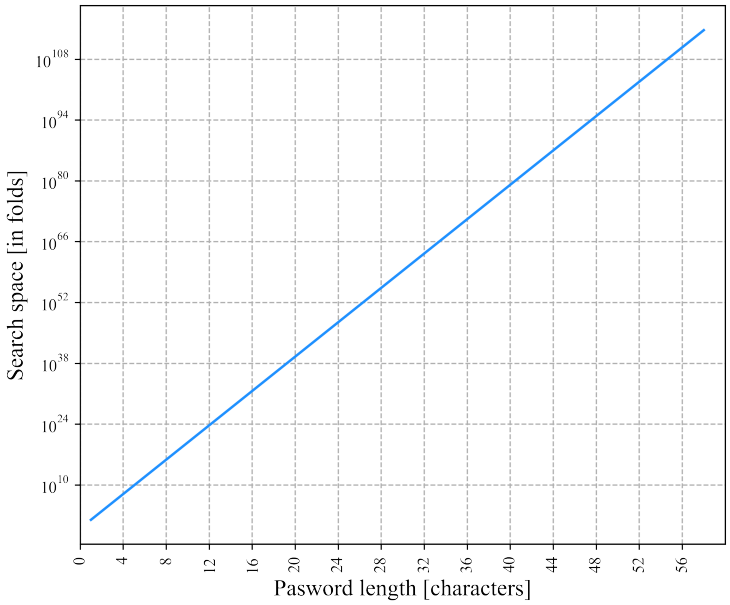


Fig. 2.4 Search space of the newly set of annonymized MAC addresses

The selected passwords can be of any type, such as ASCII characters. In order to illustrate the improvement of the search space, let's consider the ASCII printable characters, which are around 95; thus, a character will consume 8 bits. Even though the 8-bits can support up to 256 symbols, only the 95 ASCII characters are exploited for the system. Hence, when a single character is chained with a MAC address, the search space will stretch 95 times larger. Fig. 2.4 shows how the usage of ASCII character passwords quickly enlarges the search space of the MAC addresses. According to the result, if we consider the 22-byte idle slots of SHA-224 for the MAC address, the search space can be improved 10^{45} folds, hence instead of 2^{48} the new search space becomes $2^{48} \times 95^{22}$ just by chaining a password.

The other important parameter for the password is how frequently it should be used with the MAC address. As a result, if a single password of length 22 bytes is permanently stored in a SHA-224, the password itself may require up to 95^{22} brute force attempts for the ASCII characters. In this case, the search to re-identify the source MAC address is harder but not impossible since the security is bottle-necked to using a single password. This is not encouraged in the cyber security policies either, i.e., passwords should be changed on a regular basis. On the contrary, when passwords are changed frequently, their effects need to be observed, mainly on our mobility tracking. Note that, while supporting privacy according to the GDPR, the main goal is to track people's mobility via their MAC addresses. For this reason, we have analyzed the effect of password frequency on the set of MAC addresses. In the experiment, sets containing 30k, 700k, and 1.2 million unique MAC addresses were tested separately. During the test, k (i.e., $k \in \{1,2,\ldots,30\}$) passwords are applied to a set, i.e., the password is changed k times over a set, thus k is referred to as password frequency. In order to apply k passwords to a set of MAC addresses, the set is partitioned into k parts. Then, the different k passwords are chained with the different k partitions before applying the SHA digest. The data set partitioning can be based on the time of the day. For instance, the 30k MAC addresses in the first set are collected from a single day, as shown in Fig. 4.4. Thus, when the k passwords are applied to the set, it means every $m = 1440/k$ minutes a new password will be generated and be chained with the incoming probes.

As shown on Fig. 2.5, when we use more than one password, the original data set will be modified, i.e., the size of the new data set is larger than the original. Since we are focused on monitoring mobility, using passwords frequently will create bias in the head count, on top of the randomization challenge. In the mobility tracking, however, the head count is at most important on the daily bases. Thus, the password change can be performed once per day to sustain the original database as much as possible, as shown in Fig. 2.5. In order to mitigate the marginal bias, the change can be made at times when the traffic is quite insignificant, such as at midnight.

Operation cost

In this section, the performance of the anonymization process, with password chaining, is evaluated in CPU time milliseconds for the four hash functions: SHA-224,

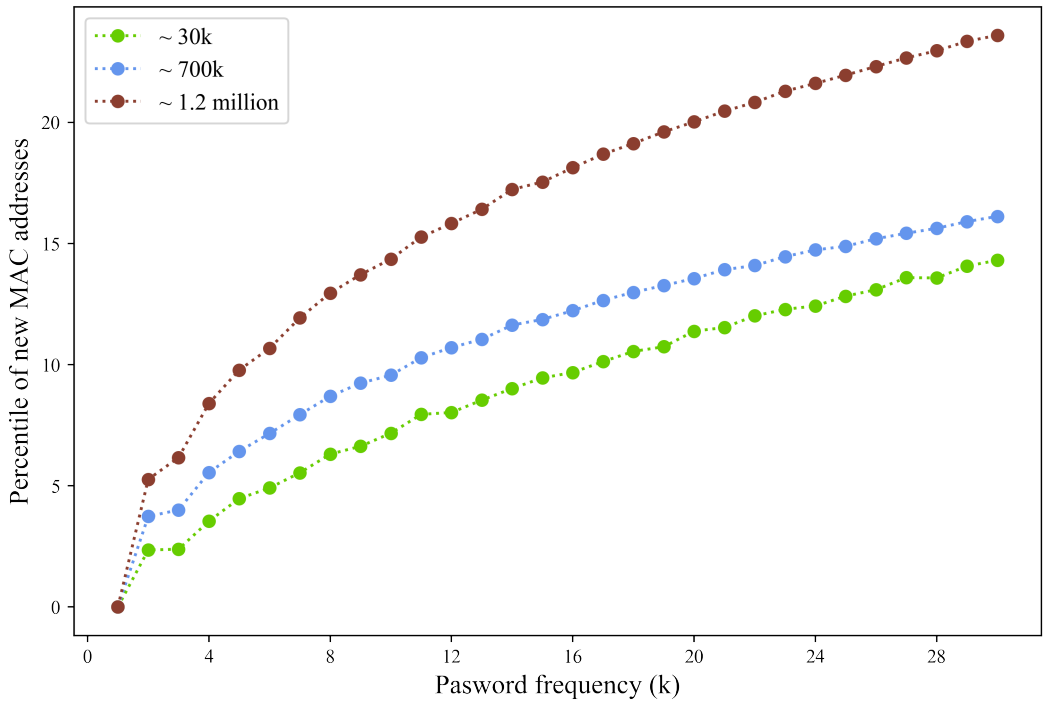


Fig. 2.5 Password application frequency

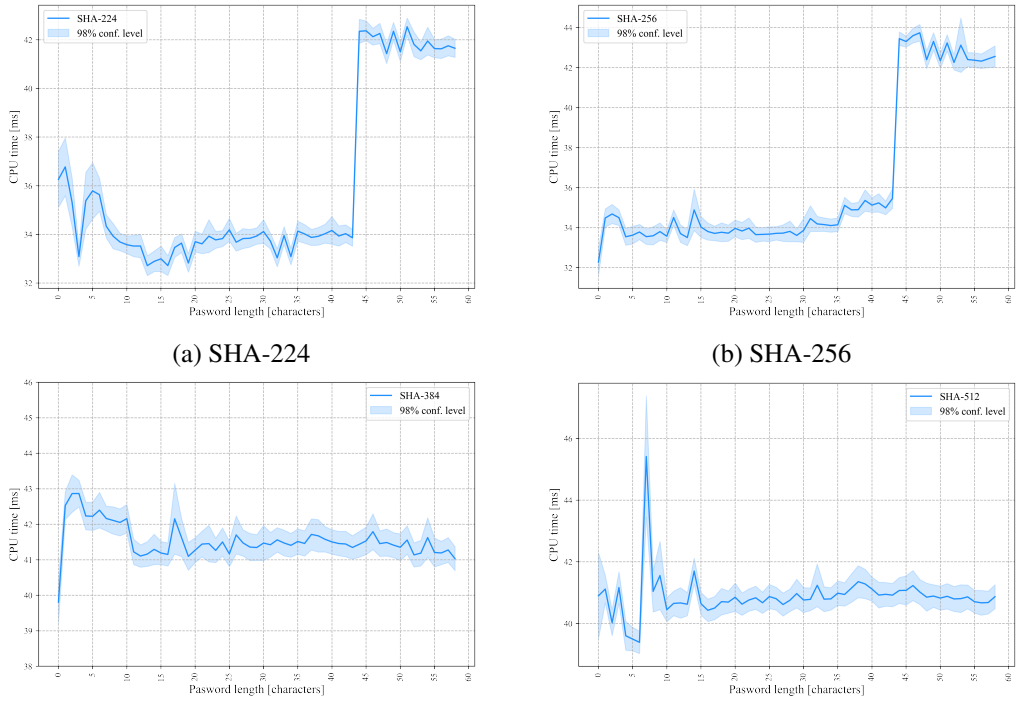
SHA-256, SHA-384, and SHA-512. For this operation, the CPU time is computed as the time it took to complete chaining 50,000 MAC addresses with a password of length d in characters and digest the chained 50,000 modified identifiers. The results are presented as follows:

SHA-224

The function takes each chained identifier and outputs a fingerprint of size 28-bytes. It took around 34 milliseconds, mostly for passwords of length less than 44 characters ($d <= 43$), with a 98% confidence interval. Instead, for larger d, $d >= 44$, it took an immediate shift to around 42 milliseconds.

SHA-256

Similar to the SHA-224, the operation took around 34 milliseconds to digest the 50,000 chained identifiers. Again, when used with longer password strings of length d greater than 43, the performance decreased quickly to 43 milliseconds. Note that the output of this hash function has length of 32 bytes irrespective of the password length d, which is closely related to the SHA-224 having only four bytes less.



(a) SHA-224 (b) SHA-256

(c) SHA-384 (d) SHA-512

Fig. 2.6 Secure anonymization CPU time

SHA-384

The chaining and digesting of 50,000 MAC addresses took mostly 41.5 milliseconds for SHA-384, the 48-byte hashing function. Compared with the previous two (SHA-224 and SHA-256), the SHA-384 took 7.5 milliseconds more. The reason for this delay is that the hash function requires 80 rounds to complete the digest, whereas SHA-224 and SHA-256 only require 64 rounds.

SHA-512

The digesting function takes similar rounds as the SHA-384 but outputs a fingerprint of size 64 bytes. In order to complete hashing the 50,000 chained MAC addresses, it required mostly 41 milliseconds as the SHA-384.

2.5 Final remark

In this chapter we have covered the foundation of a mobility tracking application. Based on two different types of WiFi signal sensors (off-the-shelf and ad-hoc designed), probe request frames were collected from mobile devices. Since the packets include personal data, in particular the MAC address of the device, personal data protection was required during the data collection and processing phase. Thus, by identifying the limitations of the currently available privacy protection techniques, the implemented privacy-preserving scheme has addressed the privacy challenges for WiFi-based mobility tracking.

Chapter 3

IoT-based Mobility Tracking for Smart City Applications

Part of the work presented in this chapter has been published in [27, 32]:

- *Kalkidan Gebru, Marco Rapelli, Riccardo Rusca, Claudio Casetti, Carla Fabiana Chiasserini, Paolo Giaccone, "Edge-based passive crowd monitoring through WiFi Beacons," Computer Communications, Volume 192, 2022, Pages 163-170, ISSN 0140-3664, https://doi.org/10.1016/j.comcom.2022.06.003.*
- *K. Gebru, C. Casetti, C. F. Chiasserini and P. Giaccone, "IoT-based Mobility Tracking for Smart City Applications," 2020 European Conference on Networks and Communications (EuCNC), 2020, pp. 326-330, doi: 10.1109/EuCNC48522.2020.9200941.*

It is widely believed that IoT systems will have a momentous impact on people's everyday lives, as testified by the development of specific use cases for upcoming 5G networks. Nowhere will this impact be more tangible than in our cities. One of the key smart city scenarios addressed by the 5G-EVE project [33] requires the identification and quantification of people in sensitive areas (e.g., for safety and security purposes, such as during large crowd gatherings) or in areas of transit (e.g., for the purpose of dimensioning transportation networks or transit/parking/sheltering infrastructure, etc.). While the detection of presence and head count is important, more valuable information would stem from the identification of flows of people. Cameras can be used for this purpose, although they require a high upfront invest-

ment, resource-consuming detection software, and maintenance, not to mention the privacy concerns they usually raise.

Alternative solutions exist, such as sensors that scan the WiFi bands and passively capture probes transmitted by smartphones as they try to identify known nearby WiFi access points. However, these sensors have some limitations: (*i*) they only detect people who have a smartphone (although it can be argued that this is now the majority of passers-by); (*ii*) if used alone, they only quantify the presence of people, not the path they are taking; and (*iii*) the information they expose is non-customizable and is heavily influenced by implementation nuances in WiFi probe timing, necessitating considerable inference.

In this work, we address the above concerns, presenting a framework that uses data collected by commercial WiFi probe-detection sensors to infer flow densities and direction of transit of people on city streets. As mentioned above, inference techniques have to contend with the implementation uncertainties and partiality of information exposed by commercial scanners. For this reason, we engaged in a measurement campaign in a real testbed scenario, realized within the 5G-EVE project, that allowed us to establish a ground truth on which to test our framework.

3.1 System architecture

At one of its site facilities, the Italian site, the 5G-EVE project includes a *safety and environment* use case for smart cities. The goal of this use case is to manage large crowds, primarily students on their daily commute. As a result, the cooperation between TIM and the Comune di Torino supports the 5G development project by supplying cellular infrastructure and Meshlium WiFi sensors, as mentioned in Sec. 2.1. The mobility monitoring system developed as part of the 5G-EVE project is based on the sensor devices and network infrastructure provided, as depicted in Fig. 3.1. The next sections detail all of the architecture's components and their functions in supporting the crowd management system.

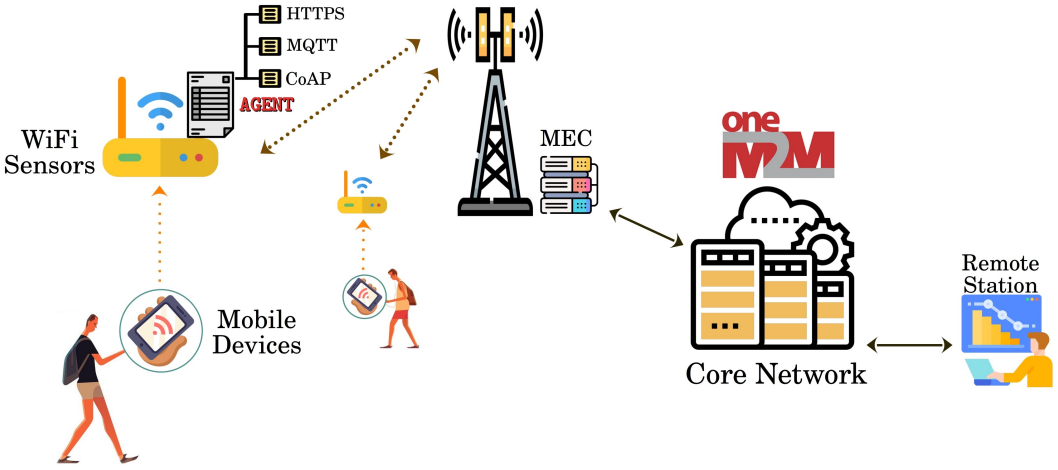


Fig. 3.1 5G-EVE architecture

Mobile Devices

Mobile devices, smartphones in particular, nowadays support different modes of communication. Among them, cellular communication and WiFi communication are the most dominant ones since they are the currently available technologies to reach the outside world for data and service at the tip of the finger. Although mobility tracking is possible through the cellular infrastructure, the information is strongly protected by the service providers as well as by law. Thus, it cannot openly support businesses and public services that are interested in mobility information. WiFi instead, not as strict as cellular communication, emits trails of signals that can be used for tracking with manageable and much lower privacy risks, i.e., without invading the privacy of the device owner.

During mobility, it is common to see people with smartphones performing personal activities such as online social communication. These smartphones, with or without the user's awareness, are capable of transmitting WiFi probes periodically, depending on vendors and applications, either trying to join a network if not connected already or performing WiFi roaming when the signal strength weakens. Thus, WiFi signal sensors are used to capture these signals for mobility tracking purposes.

WiFi Sensors

The WiFi signal sensors scan the WiFi bands at 2.4 and 5 GHz while passively listening to WiFi probe packets associated with mobile devices, including those from WiFi enabled stationary devices. Once a probe is detected, the information will be stored locally first. However, before storing the information, the MAC address of the device must be anonymized with strong hash functions for privacy reasons. Thus, on the record, only the anonymized version of the MAC address should be visible.

The scanners also have a SIM card to have communication with a cellular infrastructure. After the probes along with the anonymized MAC are saved locally, an agent of the OneM2M platform, which is located on the scanners, will forward the data through the underlying cellular network towards the OneM2M storage server located on the core network.

Agents

A OneM2M agent configured on the scanners is responsible for a scheduled data transfer. The agent sends the collected probe records towards the OneM2M server on the core network through the underling network. The data exchange operation with the platform is performed by one of the accessible protocols: HTTPs, MQTT, or CoAP.

OneM2M platform

OneM2M defines standards for Machine-to-Machine (M2M) and the IoT for interoperable frameworks. The standard uses a 3-layered model to support end-to-end services. These layers are: the application layer, the common services layer, and the underling network services layer. The application layer has an entity called an Application Entity (AE) residing on one or more nodes to implement applications. The AE, which interacts with non-OneM2M systems such as mobile devices, is responsible for collecting the probes as well as interacting with Common Service Entity (CSE) which is part of the common service layer through the Mca reference point. The CSE deals with data storing and sharing, event detection and notification, scheduling of data exchanges, and device management, and can be embedded in a gateway device (middle node) or cloud service platform to expose common service functions for other entities, such as another CSE and cloud infrastructures through

the Mcc reference point. The final and third layer, the Network Service layer, also has an entity called a Network Service Entity (NSE) which provides services from the underlying network to the CSEs. While data transfer services are handled by the underlying network between the OneM2M entities, the NSE will be responsible for managing the devices. The OneM2M platform categorizes the set of nodes into two domains in general. The first is the field domain, which includes sensors and gateways, while the second is the infrastructure domain, containing larger computer servers and applications.

In the project, the OneM2M communication between the sensors in the field domain (which contain an agent of the OneM2M) and the servers in the core network (which contains an agent of the OneM2M) is through a cellular network. The collected probes will be forwarded from the sensors to the servers periodically every two minutes by the OneM2M agent located on the sensors. The OneM2M platform is based on RESTful implementation following a store and share methodology. Based on the standard, data can be sent from an entity (e.g., sensors) called a producer and read by an entity called a consumer (e.g., a remote station). The process is performed through request/response in HTTP and publish/subscribe in MQTT.

Remote station

The remote station is the final component in the architecture. After the probe request packets are collected by the sensors and stored on the OneM2M server, they should be analyzed to identify people's mobility behaviors in the environment, thus the need for the remote station. The remote station is mainly used by privileged users to remotely access the data from the OneM2M platform using one of the three available protocols, i.e., HTTPS, MQTT, or CoAP. These privileged users can either subscribe to receive data notifications from the platform or directly download the data for the required analysis and management purposes.

3.2 Proof of concept scenario and challenges

For performing the mobility tracking activities, the two commercial WiFi sensors installed at the gates of the *Politecnico di Torino* (shown on Fig. 2.1) along with the 5G-EVE architecture (Fig. 3.1) are used for our testbed. By using the testbed,

the probe request packets were collected from the people passing-by the scanners, then the frames were stored on the OneM2M server. After that, the collected data was downloaded from the remote station for the analysis. However, there were mainly three challenges in performing the operation to achieve the objective. These challenges are from the default behaviors of the scanners, the effect of randomization, and the behavior of the environment. These challenges are explained in detail in the following sections.

3.2.1 Default behaviors of scanners

The off-the-shelf scanners used on the testbed have their own default configurations that cannot be accessed by us. One of the behaviors is that they store a limited part of the detected probe request frame. The only available attributes of the frame are the MAC address (hashed), the vendor information, the RSSI, and the timestamp. Among them, the vendor information adds no value to our mobility tracking system since it is extracted from the first three bytes of the MAC address. The RSSI is used for tracking mobility on other projects with different scenarios. It is not incorporated in our methodologies for the reasons explained in Sec. 3.2.3. The other pieces of information, MAC addresses and timestamps, are included in our methodologies, but with their own challenges. The challenges related to the MAC address are explained in Sec. 3.2.2. When we consider the timestamp, packets are recorded every 51 seconds. In other words, for 51 seconds the scanner is actively listening for the probe request frames, and then all the collected packets are time-stamped once with the exact same time reference. This approach is not convenient since it hides the correct probe transmission time, thus the presence of individuals. Besides, the processing load could become imbalanced for real-time data processors due to the 51-second idleness and accumulated load. Fig. 3.2 shows how the scanners generalize the inter-probe time every 51 seconds.

3.2.2 Randomization effect

Devices have incorporated the randomization approach, which uses temporary randomized MAC addresses rather than the global unique identifier, as a countermeasure against tracking, as specified in Sec. 2.2.2. This section demonstrates the impact of

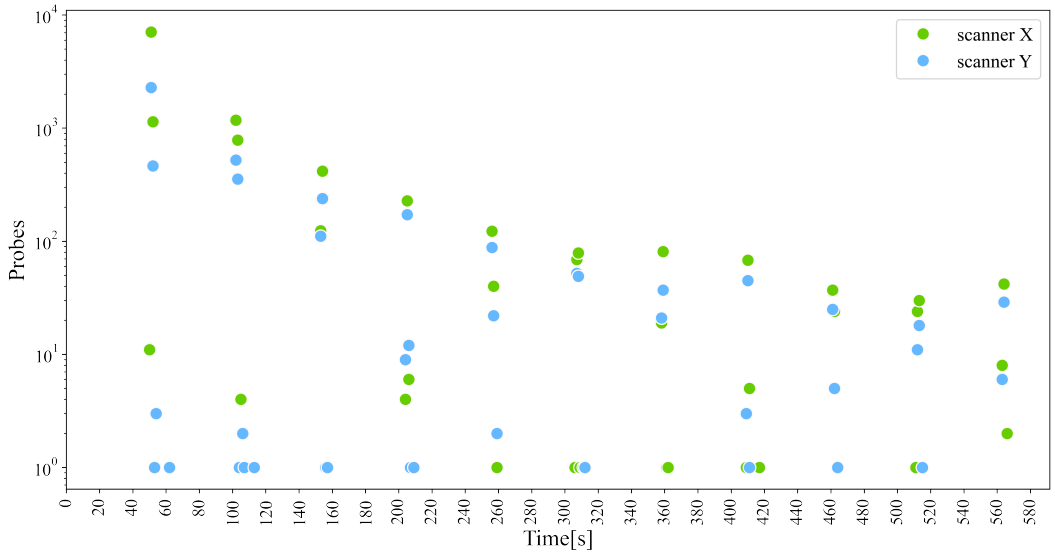


Fig. 3.2 Probe inter-times by Meshlium scanner

the gathered probe requests over a specified time period.

On a general observation, around *Politecnico di Torino*, over the period of nearly two-and-a-half months, we have found 1,260,875 unique MAC addresses. Within those days, in the second semester of 2019, the maximum number of MAC addresses seen on a single day is 30,867 on June 07, on scanner Y, as seen in Fig. 4.4. From this information, one may ask if this number is the approximate value of the maximum number of passers-by. However, this is not the case. While there are 565,550 unique mac addresses on scanner X, there are 736,150 on scanner Y (as shown in Fig. 3.3).

When we consider the number of unique devices observed in the area, which is 1.2 million, and compare it with the whole population of Turin which is less than 2 million, we cannot either state the practicability of covering more than half the population with just two scanners placed at a close distance from each other, or map a single MAC address towards a single user. Hence, the head count during mobility becomes more challenging. It is evident that the scanners are also recording the randomly generated temporary MAC address from the probe request packets.

As seen on Fig. 3.4 (a) and Fig. 3.5 (a), the number of MAC addresses that are seen for the first time over the days keeps increasing drastically, except for the Easter

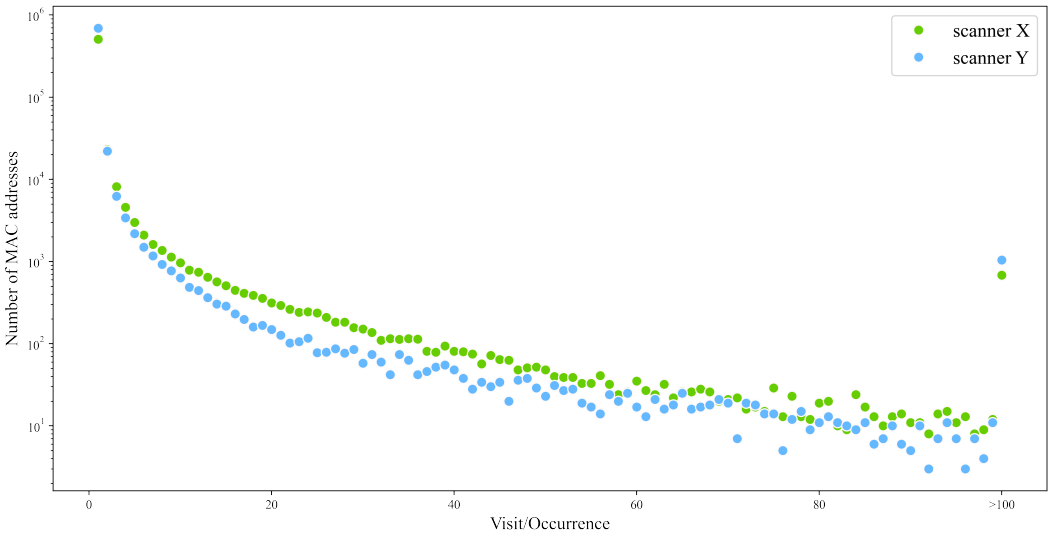
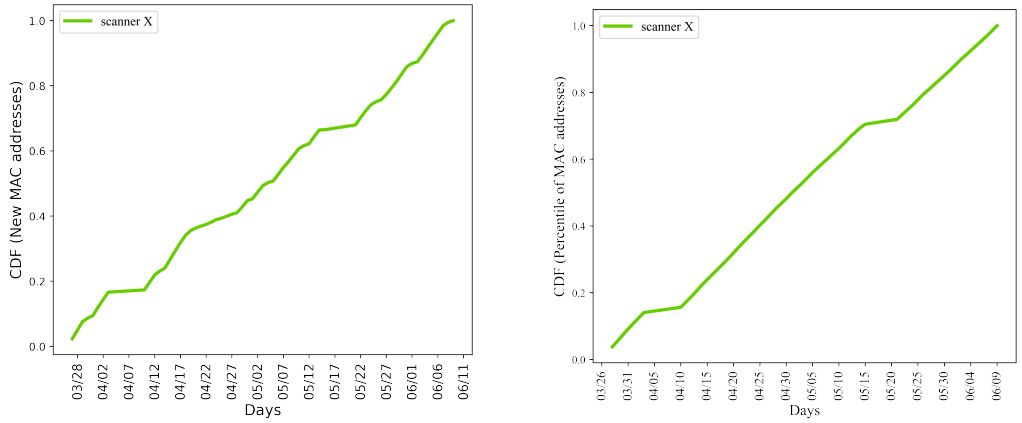


Fig. 3.3 Effect of randomization: all MAC addresses

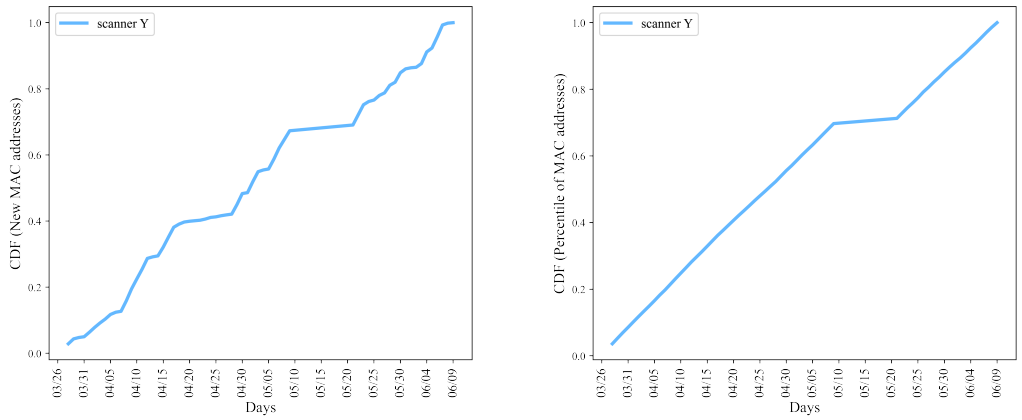


(a) New MAC addresses

(b) Percentile of new MAC addresses

Fig. 3.4 Daily new MAC addresses on scanner-X

holiday season, where the growth is lower, and when the sensors are in a power-off state, i.e., not recording. When we look at the percentile of those daily new MAC addresses, Fig. 3.4 (b) and Fig. 3.5 (b), the growth rate is linear over the period, including for the Easter holiday. That means regardless of the daily number of MAC addresses being large or small, there is a fixed average percentile that can differentiate the new addresses from those that are formerly seen addresses. Thus, the never-ending daily increase in the number of new addresses, which could be considered as the effect of randomization as well, makes crowd monitoring difficult.



(a) New MAC addresses (b) Percentile of new MAC addresses

Fig. 3.5 Daily new MAC addresses on scanner-Y

One may consider filtering devices (MAC addresses) that are seen only once over a long period of time to bypass the challenge. Even if the assumption could be correct, especially for mobility tracking because at least two points are required to trace a path, it is also impractical to conclude that all MAC addresses seen once are randomized mac addresses. Evidently, from the list of original MAC addresses which are registered at *Politecnico di Torino*, there exist a few thousand MAC addresses that are seen only once over the two-and-a-half months period, as shown in Fig. 3.6.

On the other hand, we can mostly be certain that the devices that are seen more than once are the real MAC addresses of devices. Analytically, the probability of a single randomized address being generated by more than one device in the same time frame is (let's call it $P(collision)$):

$$P(collision) = \frac{1}{2(2^n)}$$

where n is the number of bits considered for the randomization. When all 48 bits of the MAC address are considered for randomization, i.e., full-address randomization, the possibility of collision between randomized addresses ($P(collision)$) is extremely low. However, n can be lower than 48 since there are nearly 2^{15} vendors as mentioned in Sec. 2.3.1. When vendors only randomize the last three bytes of the MAC address, the maximum probability $P(collision)$ is:

$$P(collision) = \frac{1}{2(2^{24})}$$

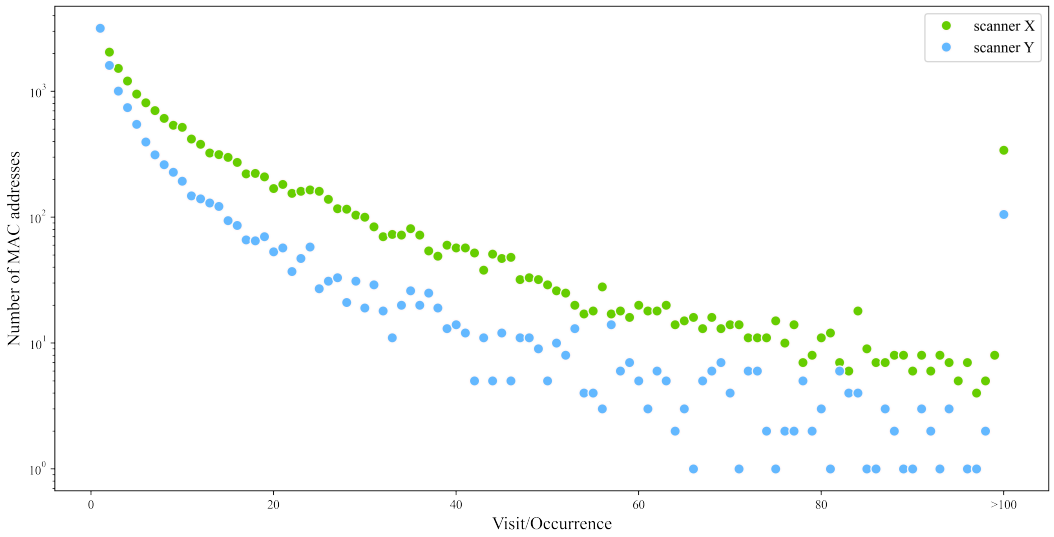


Fig. 3.6 Non-randomized Polito MAC addresses

Thus, when MAC addresses are classified between randomized and original, all the MAC addresses recorded by the scanner that are seen more than once can be categorized as original MAC addresses. Another methodology to bypass the randomization is explained in [15]. However, the work in [15] requires the sequence number and the tagged parameters of the probe request frames, which in this scenario are missing. The de-randomization methodology is adapted in Sec. 2.3.2 for ad-hoc designed sensors.

3.2.3 Environmental behavior

During the reception of the WiFi probes, the sensors can evaluate the RSSI and infer the distance from the device [34]. Thus, from the measurement, movements can be tracked. This methodology works well for indoor scenarios but not so well for outdoor scenarios because signal propagation and attenuation are heavily influenced by the environment [35]. The multipath effect and the way users hold the mobile devices also affect the received signal strength parameter and can lead to noisy signals [36]. However, in order to be certain of the possibility of RSSI based tracking, a preliminary test was performed on the testbed using two android-based Samsung and Asus smartphones.

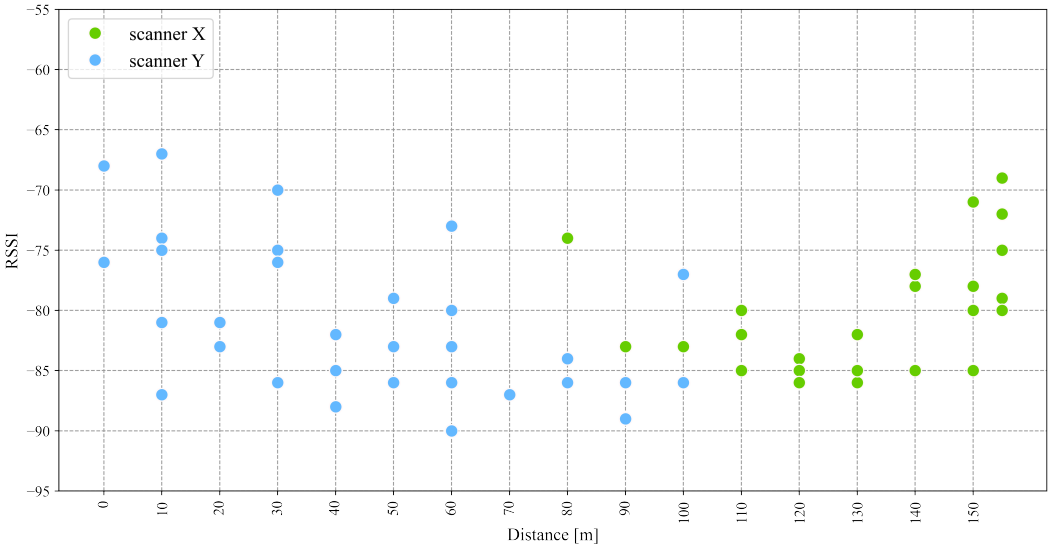


Fig. 3.7 RSSI mobility tracking: phone A

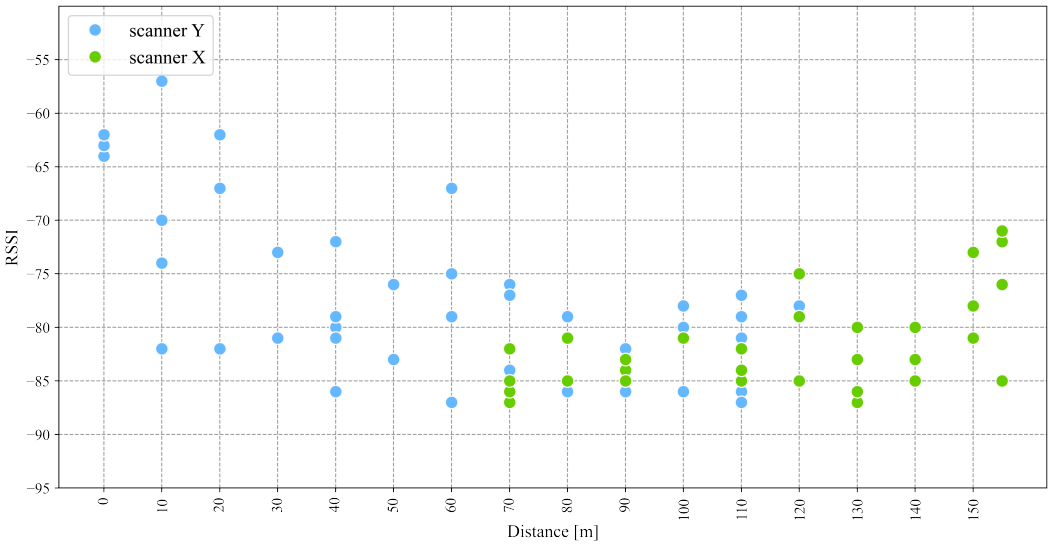


Fig. 3.8 RSSI mobility tracking: phone B

In the experiment, we walked starting from the position of scanner-*Y* towards scanner-*X*. When walking, there is a minimum of a 3 minute stop at every 10 meter distance in order to transmit at least 3 probes from the smartphones at each position. The probes from the trial were detected by the scanners. However, the collected information is different between the phones. Firstly, when we look at how far probes can be received by the scanners, scanner-*Y* has detected the probes from smartphone-*A* up to 100 meters, while it was 120 meters for smartphone-*B*. Furthermore, the number

of received probes has variations from one to five among the positions, which could translate to loss of signal in a noisy environment.

Most importantly, however, the distance measured by the scanner is unreliable since there is a clear disparity between the distance estimations from the same position. For reasons mentioned so far, the RSSI will not be considered to track the mobility of our outdoor tests.

3.3 Mobility flow tracking

The purpose of this work is to track people's movements in the environment. For this reason, we have used the 5G-EVE architecture and testbed (Fig. 3.1). On the testbed, we have carried out mobility experiments to apply mobility tracking methodologies and test the performance. In the following sections, the applied methodologies and the experiments are presented in detail.

3.3.1 Mobility pattern detection

In this section, we will provide an assessment of mobility patterns based on the two scanners. Our main goals here are, first, to identify the most popular mobility patterns; second, to observe the phenomenon of MAC randomization on the patterns; and finally, to investigate whether the methodology for identifying the patterns is sufficient or not for a mobility tracking system.

We downloaded a data set from the OneM2M server for the analysis, a typical week in October 2019. The downloaded data set consists of 195,762 distinct MAC addresses. It is expected, among the unique set of addresses, for devices to appear with more than one MAC address, as discussed in the concept of randomization. Hence, we can only infer an upper bound on the number of devices near-by or passing-by the two scanners.

In order to make a classification between the mobility patterns taken by the passing-by individuals, we have come up with a methodology to represent the coverage informa-

tion with strings. Fortunately, the scanners have been identified and conferred with the strings 'X' and 'Y' thus far, so there is no need to change the representations. Therefore, the testbed area which is covered by scanner X only will be labeled as "X", and the area covered by scanner Y only will be labeled as "Y" during the process. In addition, as seen in the study of RSSI tracking with the two mobile devices (Fig. 3.7 and Fig. 3.8), there is an overlapping area that is covered by both scanners. Thus, besides the strings 'X' and 'Y', a third string is required to represent the overlapping area to have better precision. Whenever a device is seen on both scanners at the same time, therefore, it will be assigned the string 'Z'.

For each collected MAC on the analyzed trace, we computed the temporal sequence τ of detection events that can be represented as follows: $\tau = [(t_i, s_i)]_i$, for increasing values of t_i, $(i = 0, 1, 2, ...)$. A generic pair in τ represents the events according to which scanner s_i detected the device at time t_i , where $s_i \in \{X, Y\}$.

The next step after labeling the coverage area is to set up a considerable time frame for a single movement between two positions, i.e., to determine a maximum time limit for the movement between the possible coverage areas. For this reason, on the test bed, 4 minutes is assumed to be sufficient to change between coverage areas during mobility. Thus, τ is partitioned into sub-sequences by gathering all the consecutive coverage events occurring with a time difference of no more than 4 minutes.

According to the results shown on Fig. 3.9, most of the devices are seen under a single coverage area 'X', 'Y', and 'Z', which means the movement is either limited or the device is stationed. In order to observe the effect of MAC randomization, the collected list of 34,927 *Politecnico* MAC addresses is incorporated into the result as well. Note that these MAC addresses are not randomized since they are collected after being associated with one of the access points at the *Politecnico di Torino*.

Referring to mobility as a transition between the set of areas, it is practical to classify all the mobility patterns having more than one string, such as 'XY' and 'YX', on the labels. It is also possible to remark that these mobilities are not affected by the randomization process. That is to say, while we consider the occurrences of the

Fig. 3.9 Most popular mobility patters

patterns '*X*', '*Y*' and '*Z*' as an upper bound due to randomization, the occurrences of the patterns such as 'XY' and 'YX' can be considered as a fairly exact number. The preceding statement can be justified in two cases. The first one is to take into account the probability of having two mobiles using the same MAC address at the same time, i.e., a collision during the randomization, which is extremely low as explained in Sec. 3.2.2. Thus, when a single MAC address is captured by two scanners in a different but limited time frame, unlike '*Z*' where a signal is detected by both scanners at the same time, we can say the probes are from the same device. On the contrary, however, the device could be in its randomized state during the 4 minute time frame mobility, which leads to the second required justification. Since we are not interested in the device's identity, whether it is randomized or not, it cannot affect the number of devices for that time frame. Therefore, the head count during the mobility can be supported by having more than one scanner covering different areas, regardless of the randomization effect.

Although the string-based mobility pattern classification provides a solid foundation for tracking groups and individuals, it is limited in another way. The methodology could support the classification of directional flows at a higher level, but not in details. We can consider a scenario where there is more than one route in the same direction. For instance, two individuals, starting from the same position, may take different

paths to reach the same destination, where in this case, the mobility pattern classifier can not separate the two flows. Therefore, another methodology, the foot-printing technique, has been developed for classifying paths, including flows in the same directions.

3.3.2 Foot-printing technique for mobility tracking

The aim of our mobility tracking system is to associate the probes transmitted by a mobile device and detected by the WiFi sensors to the most likely path, across a given set of predefined paths that are monitored in the area. The classification is based on some preliminary experiments to build the "ground-truth" information, which allows to build a catalog of footprint vectors for each possible path. Thus, when a new mobile device is detected by the sensors, the sequence of coverage events detected by the scanners is compared with all known footprints and the path with the most similar footprint is associated as output of the mobility tracking, as detailed more formally in the following.

Let $\mathscr{P}$ be the set of predefined paths in the considered area to monitor and let $p \in \mathscr{P}$ be a generic path. We assume to have just 2 WiFi sensors, denoted as X and Y. In order to compute the footprint f_p of a path p, we let the sensors collect probe samples by having a person walk along p for k times, carrying a device. In the following, we will refer to such a device as "ground-truth device" and to each walk along p as a "run". Each run generates a temporal sequence $\mathscr{T}$ of detection events that can be represented as follows: $\mathscr{T} = [(t_i, s_i)]_i$, for increasing values of t_i, $i = 0, 1, 2, \ldots$. A generic pair in $\mathscr{T}$ represents the events according to which sensor s_i detected the ground-truth device at time t_i, where $s_i \in \{X, Y\}$. Consider the following simple example (assuming all times expressed in seconds):

$$\mathscr{T} = [(0,X),(30,X),(60,X),(90,X),(100,Y),\\ (120,X),(130,Y),(160,Y),(190,Y),(220,X)].$$

The above expression can be interpreted in the following way: the ground-truth device was detected by scanner X at times $0, 30, 60, 90, 120, 210$ and by scanner Y at times $100, 130, 160, 190$. Note that detection events occur at multiples of 30s,

i.e., periodically as in the considered off-the-shelf scanners (see Sec. 2.3.1), and the sampling events have a 10s offset between scanners. Now from $\mathscr{T}$ we compute a *path map* $\gamma(t_i) = 2$ if $s_i = X$ and $\gamma(t_i) = 1$ is $s_i = Y$. These two integer values have been arbitrarily chosen and do not affect at all the final classification result. Fig. 3.10 shows the path map for the considered simple case scenario.

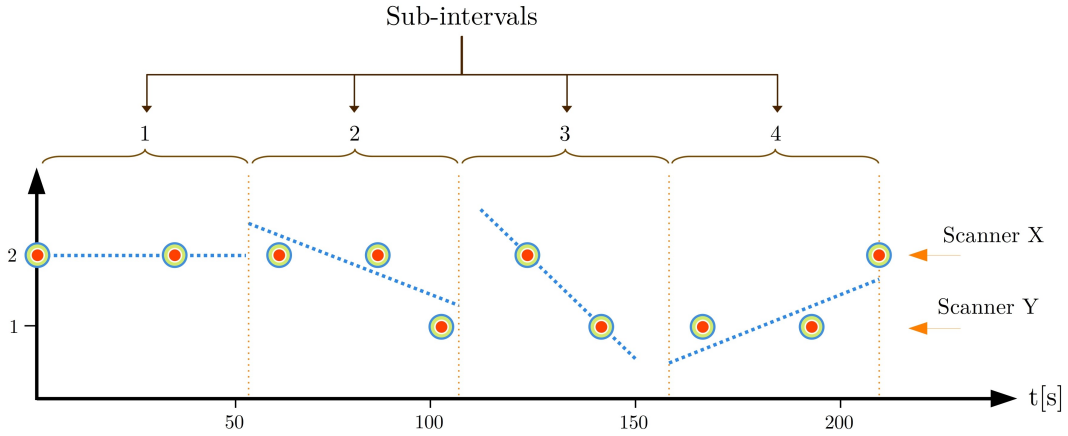


Fig. 3.10 Sample path map of the simple scenario

Let δ be the observation period, i.e., the total duration during which the ground-truth device has been detected, i.e., $\delta = \max_i\{t_i\} - \min_i\{t_i\}$. Let us now partition the observation period into N temporal sub-intervals, each of duration δ/N. Notably, N is the only parameter that should be tuned according to the proposed scheme and later we will show that already $N = 4$ yields good results. In the example, $\delta = 210$s and each sub-interval lasts 52.5s when $N = 4$.

With the above data we can now compute the footprint f_p. We remark that this is just one of the possible footprints that can be designed for path identification. The footprint we use is represented by a vector of $2N$ real numbers, formally $f_p \in \mathbb{R}^{2N}$. We divide such a vector in two parts:

- *coverage part*: the first N values of the footprint ($f_p(i)$ for $i \in \{1,\ldots,N\}$) are computed as the average of $\gamma(t)$ for each sub-interval. This weighs the detection of the device from multiple scanners during the same intervals.

- *direction part*: the last N values of the footprint ($f_p(i)$ for $i \in \{N+1,\ldots,2N\}$) model the mobility direction between the two scanners for each sub-intervals. Formally, it is computed as the slope of the best fitting linear interpolating function of the samples within the considered sub-interval.

In the considered example, the sub-intervals would be:

$$[0,52.5),[52.5,105),[105,157.5),[157.5,210]$$

and the corresponding foot-print would be computed as:

$$f_p = [\underbrace{2,1.67,1.5,1.33}_{\text{coverage}},\underbrace{0,-0.019,-0.1,0.018}_{\text{direction}}]$$

Indeed, during the first sub-interval the ground-truth device was detected by only by scanner X (i.e., 2) and the corresponding slope is 0. During the second sub-interval, it was detected twice by X (i.e., 2) and once by Y (i.e., 1), thus the average is 1.67 and the corresponding slope is negative, suggesting that the device moved mainly from X to Y. A similar reasoning applies to the following two sub-intervals.

By performing many runs with the ground-truth device, a set of footprints is attached to each path. Thus, in order to find a match for a new device, the mobility tracking system computes its footprint and looks up the most similar footprint, using a simple Euclidean norm to evaluate the distance between vectors. In case many paths show footprints at a minimum distance, the path with the maximum number of minimum distance footprints is chosen. If still more than one path is found, the device is marked as untraceable.

3.3.3 Ground-truth experiment

To put our footprint methodology to the test, we devised a mobility scenario in which four paths denoted by AB, BA, AC, and CA are chosen for the experiment, as shown in Fig. 3.11. The letters A, B, and C on the map (Fig. 3.11) represent the starting and ending positions during the ground-truth data collection process. The paths are significant for the people accessing services from the area as well as the people residing on the blocks. The mobility tracking, especially path detection, is challenging since

all the mentioned paths are partially covered by the two scanners. In the experiment, 17 runs were performed for each path at a slow pace. As a ground-truth device, a Samsung A6 smartphone with Android 9.0 was used to broadcast probes during the mobility. In order to impose frequent probe transmissions during the walk, the smartphone was forced to detect all available WiFi networks in the area. During the walks, the actual time at which the walk started and ended has been recorded for each run on the paths.

The scanners were also collecting the detected probe packets from the phone and storing the extracted information on the OneM2M server. Once after completing the activity, all the trace logs related to the ground-truth device, referring to the period of interest, are downloaded from the OneM2M server. Then, all the runs were foot-printed with the corresponding temporal information according to the foot-printing methodology explained in the previous section. From these sets of footprints, a catalog is formed for later path classification and performance testing.

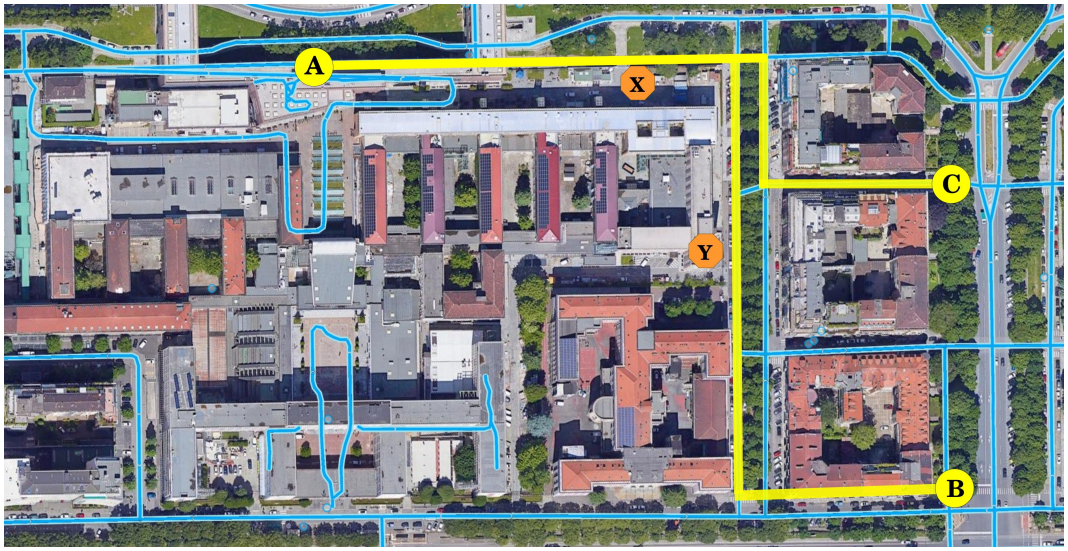
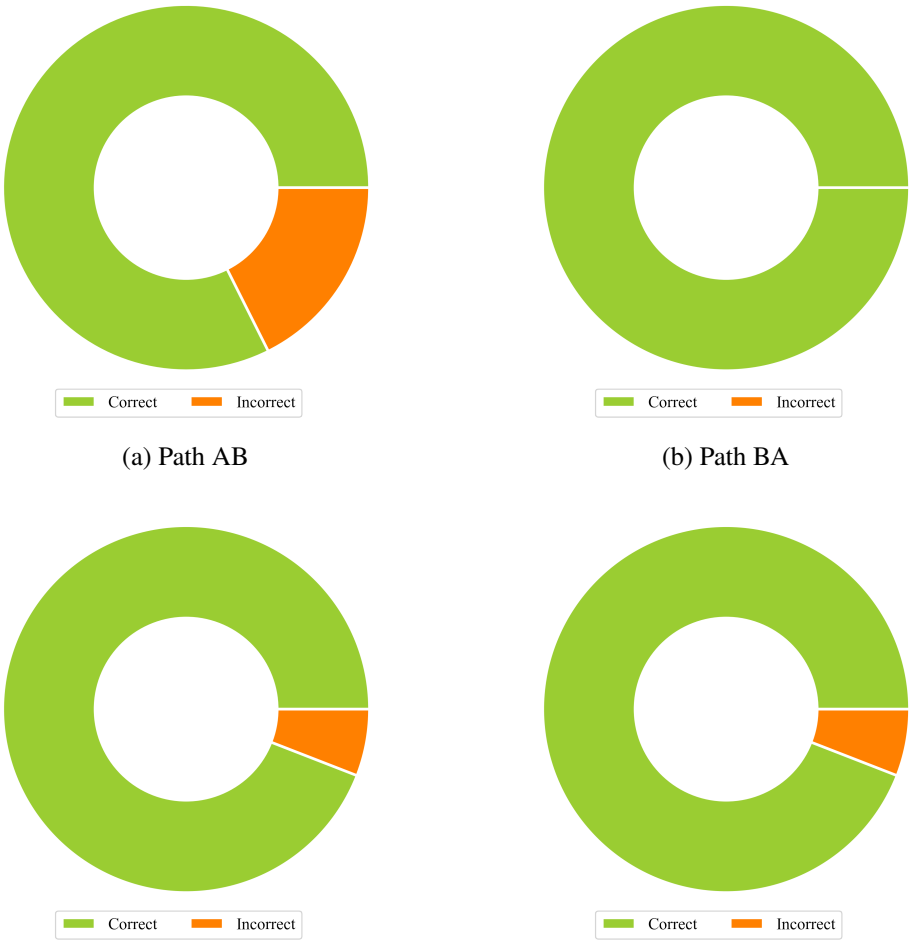


Fig. 3.11 Paths for ground-truth

Fig. 3.11 shows the four paths (AB, BA, AC, and CA) along with the three starting and ending points (A, B, and C), marked with yellow lines and circles. The two scanners labeled as X and Y are also presented with the orange hexagons.

Experimental results

In order to evaluate the performance of the mobility tracking methodology, the accuracy of the path classification was tested using a cross-validation technique. During this trial, a single run was used as a test data while the remaining runs were used as a training set. According to the result, most of the test runs were correctly identified by the approach. However, minor incorrect classifications were also detected from the overall test runs of the paths, as shown in Fig. 3.12. Note that these errors are due to the strong similarity of two paths that are in the same direction.



(a) Path AB (b) Path BA

(c) Path AC (d) Path CA

Fig. 3.12 Accuracy of mobility tracking algorithm

3.4 Final remarks

The focus of this chapter was the WiFi-based mobility tracking system. We have used the 5G-EVE architecture for our testbed, collecting probe request frames through off-the-shelf WiFi sensor. Through our tracking methodologies, we were able to classify the most popular patterns as well as mobility paths. We have validated our approach through ground-truth experiments.

Chapter 4

Passive Crowd Monitoring

We made environmental observations using two off-the-shelf WiFi sensors in this section of the work. As employed so far, the two WiFi scanners are located at the gates of the Politecnico di Torino, as shown in 2.1 with X and Y representations. Since these sensors are installed on campus, a statistical analysis of the Politecnico community can be performed. Our analysis includes identifying groups from the crowd as well as detecting events from daily activities. Furthermore, our investigation is supported by a model.

4.1 Statistical analysis

The two scanners at the Politecnico have been actively collecting probe request frames since the early 2019. In order to make the observations, we have downloaded traces of data from a remote station. The downloaded data set is from the second semester of 2019, from March the 27^{th} up to 9^{th} of June. Based on that, we have made a general analysis of the environmental activities nearby the two sensors.

4.1.1 Probes

In the first phase, a general statistical analysis of probes from the two scanners over the mentioned period is presented. According to the daily statistics, there are days when the scanners record up to 54K probe packets, as seen in Fig. 4.2. On the contrary, there are also continuous days where there are no probe packets recorded

by the scanners; two rounds for scanner X, and only one larger round for scanner Y, as seen in Fig. 4.1 and Fig. 4.2, respectively. Those days with no records of probe packets are the days when the scanners were turned off since the launch was in a trial phase.

Another observation from the shapes of the probe graphs is the scale variation between the medium and large number of daily probes. For instance, on both scanner X and scanner Y, the last ten-to-twelve days of April are smaller than the most. The reason behind that was the Easter holiday season, where most students travel to join their families for the celebration.

One more phenomenon worth mentioning could be the two-day drop after most large records in the probe packets, for instance, the last two days on both the scanners and similar ones. Clearly, these are patterns showing the weekdays and weekends. While the number of recorded probe packets on the weekdays is normally larger, the weekends, especially Sundays, are smaller.

Finally, when we compare the probe records between the scanners themselves, scanner Y mostly detects a greater number of probes than scanner X. If we consider maximums, scanner Y recorded 53,922 on the 4th of June, while this number was 40,218 for scanner X on the same date. From this and the general intuitive graph observation, we can infer the presence of more activities nearby scanner Y on active days. However, better routines are observed more on scanner X.

4.1.2 MAC addresses

Every probe packet received by the scanners has a source MAC address linked to the transmitting device. On an ideal assumption, the MAC address can be used for a head count and, furthermore, for tracking mobility. With that in mind, we can look at the statistics on devices, and thus the number of people nearby or passing-by the scanners, from the MAC address analysis.

Fig. 4.3 and Fig. 4.4 show the number of MAC addresses recorded per day over the presented period. The first observation on the graph is its shape over the days, which

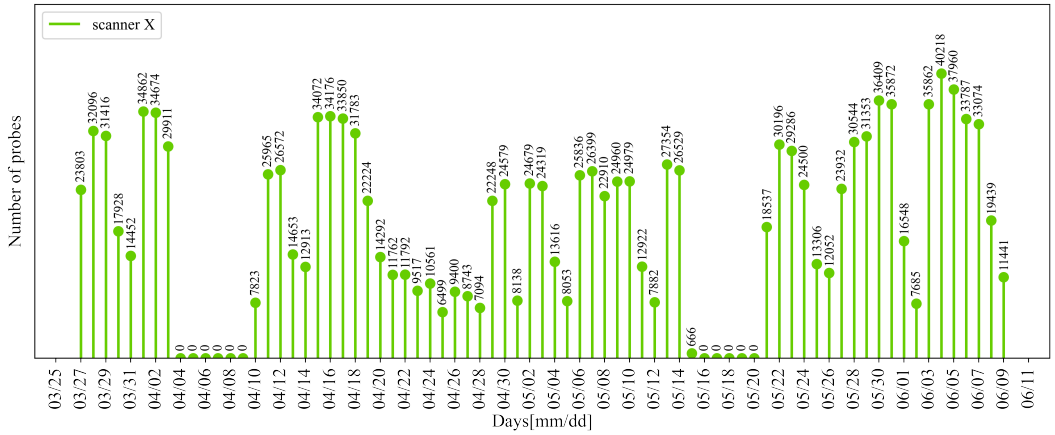


Fig. 4.1 Number of daily probes for the first scanner

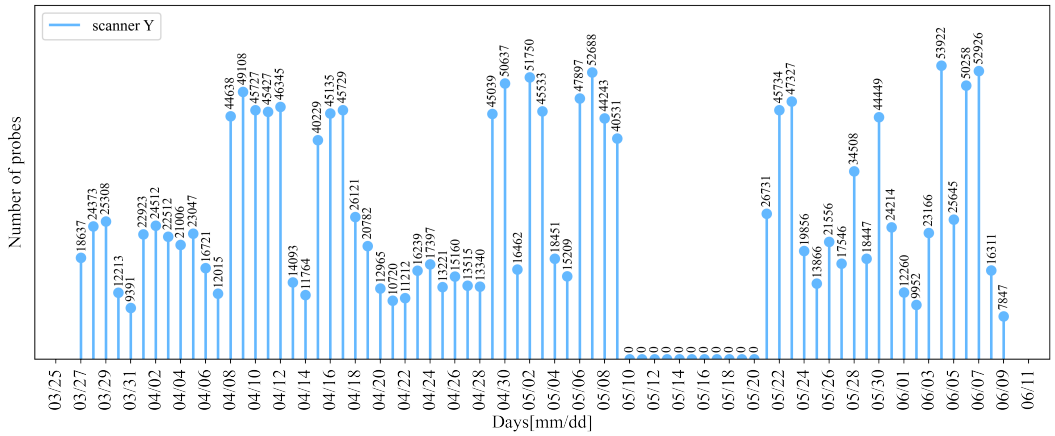


Fig. 4.2 Number of daily probes for the second scanner

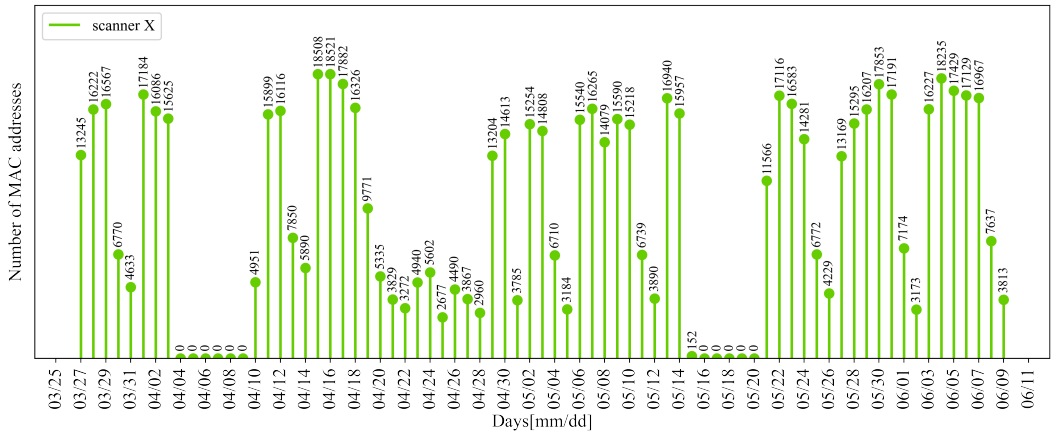


Fig. 4.3 Number of daily MAC addresses for the first scanner

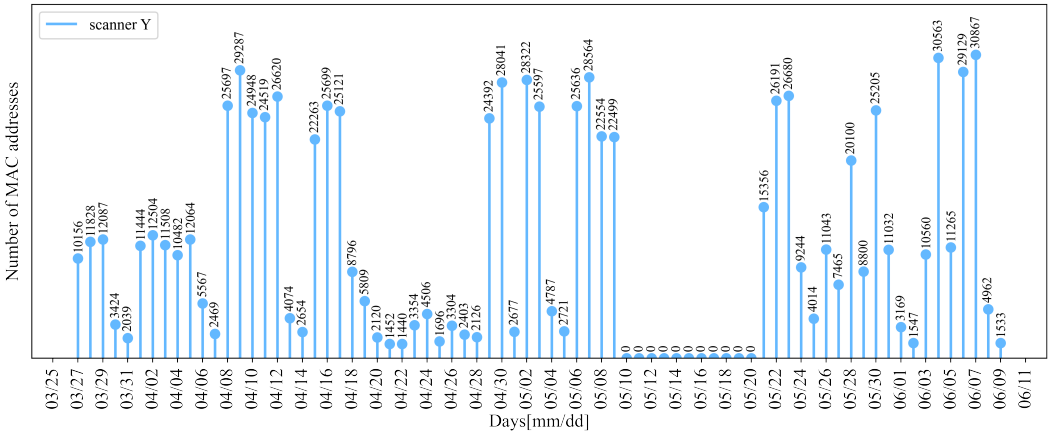


Fig. 4.4 Number of daily MAC addresses for the second scanner

complements the pattern of the probe packets presented in the probe statistics. Like the probes, we can notice the difference between the weekdays and weekends, the Easter holiday season, and the activities which are more regular on scanner X than on scanner Y. When we look at the maximum number of devices recorded by the scanner, scanner Y has detected the probes from more devices than scanner X, 30k and 18k respectively, as expected from the probe statistics.

4.2 Model based event analysis

During mobility monitoring, one of the important pieces of information for the mobility-based service providers is knowing the crowd movements. In a large crowd, there are groups and individuals performing their daily activities at different times, for varying durations. The service providers could also be interested in service load and idleness along with the time in order to improve the service quality. Therefore, in this part of the work, we have considered modeling the events in the environment with a queue system.

For the modeled queue system, we have made preliminary definitions from the queue model terms as follows. Whenever an individual or a group is seen for the first time, the event can be referred to as an arrival, and when the same individual or group leaves the area under coverage, it can be referred to as a departure. Based on arrival and departure events, the time between the arrival and the departure is considered a

service time. Thus, in this section, we will perform classifications in order to capture groups from the crowd and, furthermore, process the information to track groups, i.e., the times for the arrival and departure of a group.

On the premises of the queue model, three arrival and departure event cases can be analyzed: under scanner-X only, scanner-Y only, and under both scanners together, scanner-$X\&Y$, as shown in Fig. 4.5, Fig. 4.10, and Fig. 4.15, respectively. Note that, on the third case, scanner-$X\&Y$, order of arrival will not be considered, i.e., $X \rightarrow Y$ and $Y \rightarrow X$ are overlooked. In addition, any back and forth movement between the scanners is considered part of the service until a complete departure from both the scanners is noticed. Furthermore, on the events, when a probe is detected from a device only once but never on that day, it is considered an immediate loss after an arrival, and for a MAC address associated with more than one probe, the first probe is marked as an arrival event while the last one is taken as a departure event.

4.2.1 Scanner-X events

In this case, there are three types of events: arrival, departure, and loss. Between these events, two situations can occur; either scanner-X detects a single probe from a device over the course of the day, in which case it is assumed lost, thus marked as a loss event; or the scanner receives more than one probe, where the first probe of the device is tagged as an arrival and the last probe of the day is tagged as a departure event. The time between the arrival and the departure is labeled as the service time where the device could be transmitting more probe packets or be silent, i.e., out of scanner-X's reach.

When we look at the arrivals and departures for scanner-X on Fig. 4.6 (a) and Fig. 4.6 (b), there are two phases of growth over time. The arrival rate, initially, is much slower, with almost no activity for the first seven hours after midnight. This is expected since the campus is closed during those hours. Then, in the second phase, from 7 until 21, the growth curve went up significantly, highlighting the increase in the number of devices in the area over the period. Finally, it starts leveling off through the night after 21. On the result, the departure is at a close rate to the arrival. This shows the fact that the detected devices are leaving the site quickly. We can

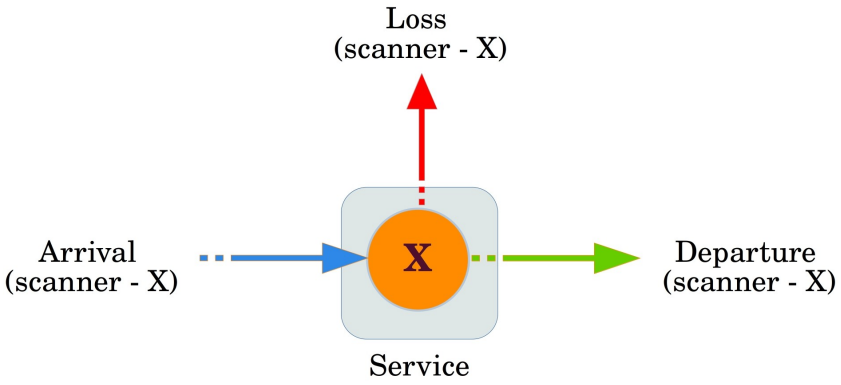
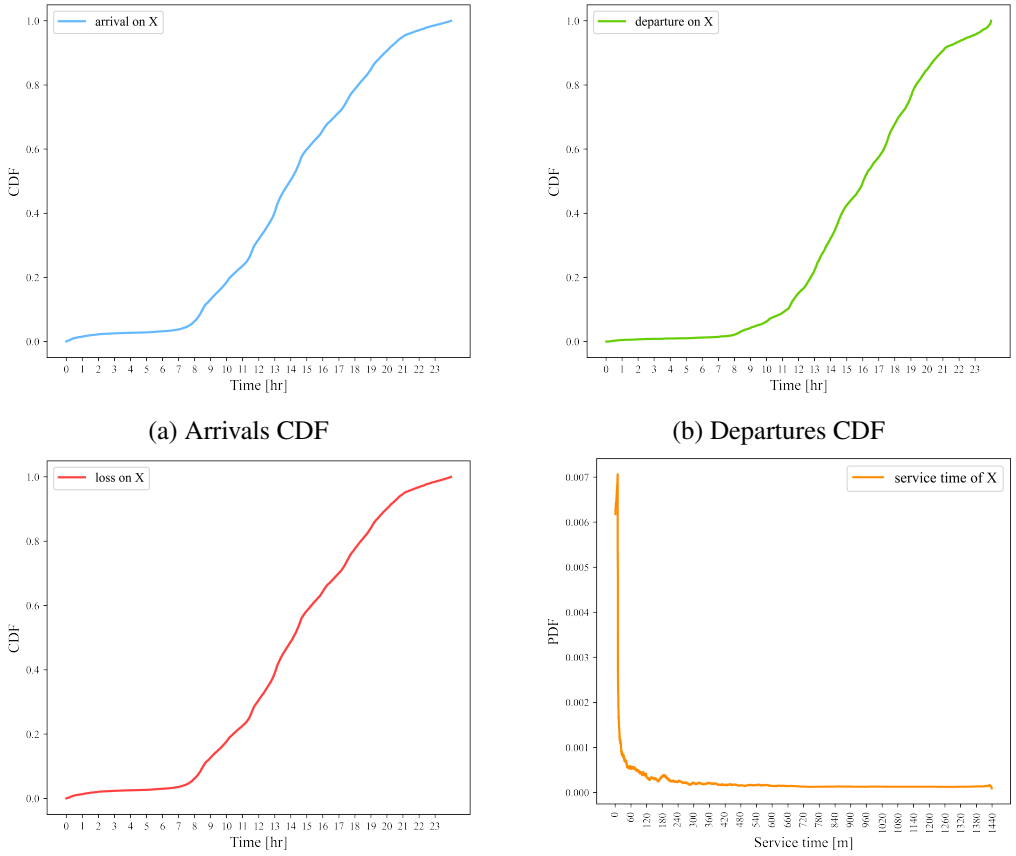


Fig. 4.5 Queue model for scanner-X



(a) Arrivals CDF

(b) Departures CDF

(c) Losses CDF

(d) Service time PDF

Fig. 4.6 Distribution of events on scanner-X

also see in Fig. 4.6 (c), that the loss curve has a similar shape to the arrival curve. This is expected since losses occur immediately after the arrival, according to our assumption. Thus, we can classify the active hours using the arrival and departure CDF distribution of the model.

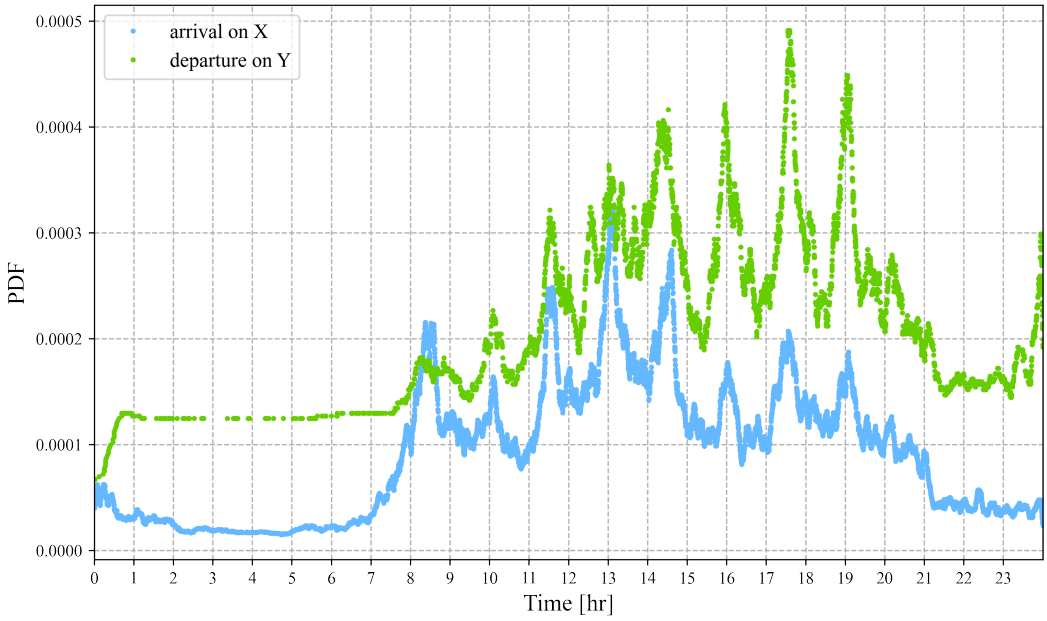


Fig. 4.7 Arrival and departure PDF for scanner-X

On Fig. 4.6 (d), we can observe the true service time distribution for the devices. According to the result, most of the devices stayed for a very brief amount of time, for a few minutes. There are, of course, devices having a service time duration of up to 3 hours or more, but they are relatively very small in number as seen from the distribution.

Finally, for the case, the PDF of arrival and departure over the times is presented in Fig. 4.7. The result shows some peak values for specific hours, revealing significant traffic during mobility. These hours are the times for the class schedules of the *Politecnico di Torino*, i.e., starting from 8:30 students either attend or leave every 90 minutes. Thus, by looking at peaks in the PDF of the model output, it is possible to detect important schedules affecting the traffic flow.

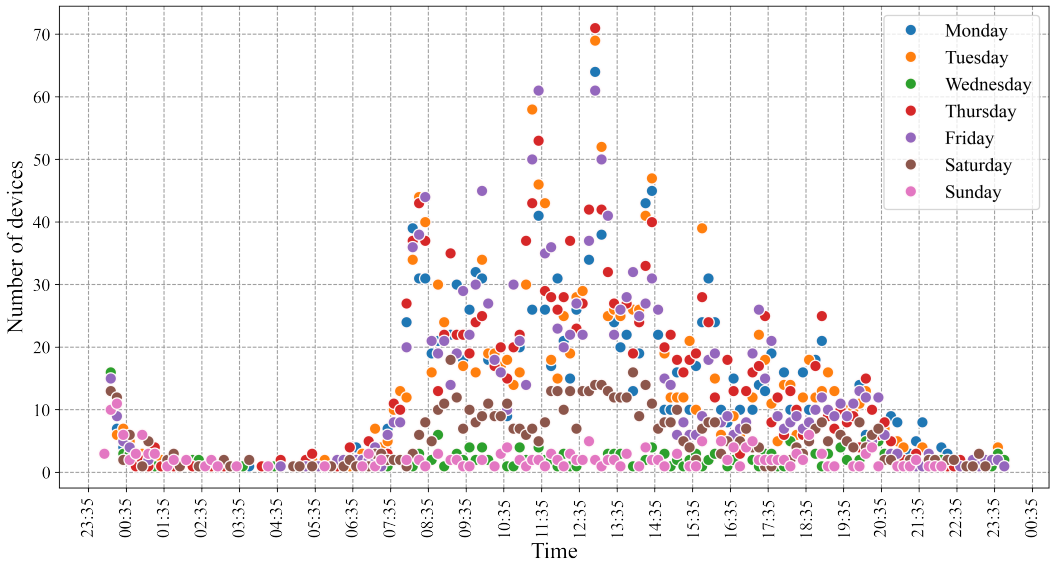


Fig. 4.8 Arrivals on scanner X

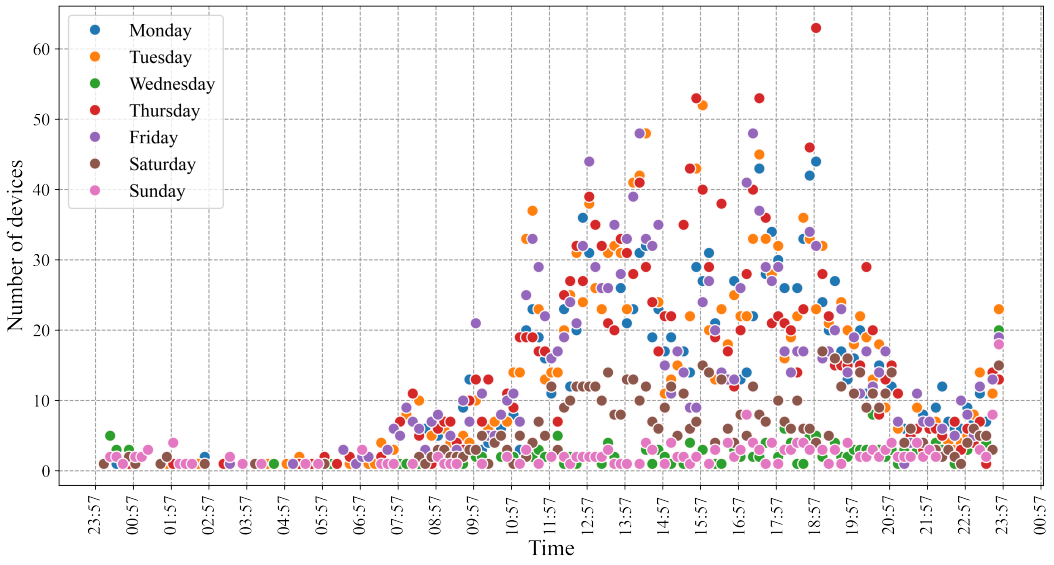


Fig. 4.9 Departures on scanner X

In order to verify the founding, we have taken daily traces for a week, from Monday 29/04 to Sunday 05/06 of the year 2019. Each circle, colored for the days, is the group of devices detected by scanner-*X* over a ten-minute window in Fig. 4.8 and Fig. 4.9. When we look the peak arrival times for the classified groups, as in Fig. 4.8, it matches exactly the PDF of the arrivals in Fig. 4.7, and similarly for the departures in Fig. 4.9 and Fig. 4.7.

4.2.2 Scanner-*Y* events

The second case is similar to the first, but instead of scanner-*X*, the events are on scanner-*Y*. The arrival and departure curves in Fig. 4.11 show a steady state for the first seven hours of the day, and then start taking off exponentially until they reach the night hours of the day. However, a small decrease in the growth rate is noticed at the 14th hour, which could indicate a meaningful phenomenon during the day that needs further observation from another point of view to see the details. Let's call the phenomenon P_1 to verify it later.

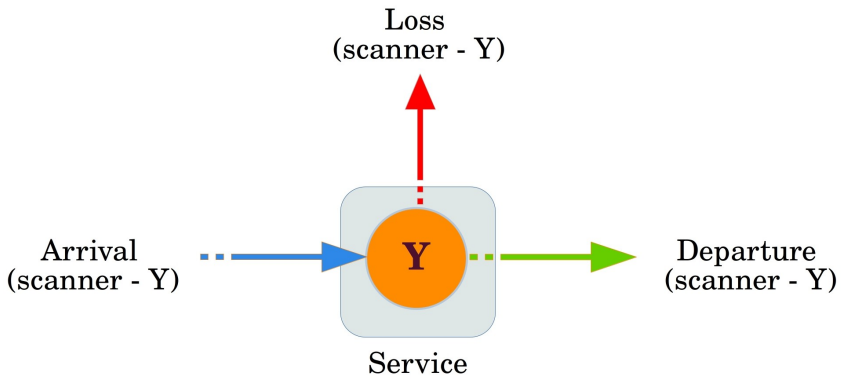
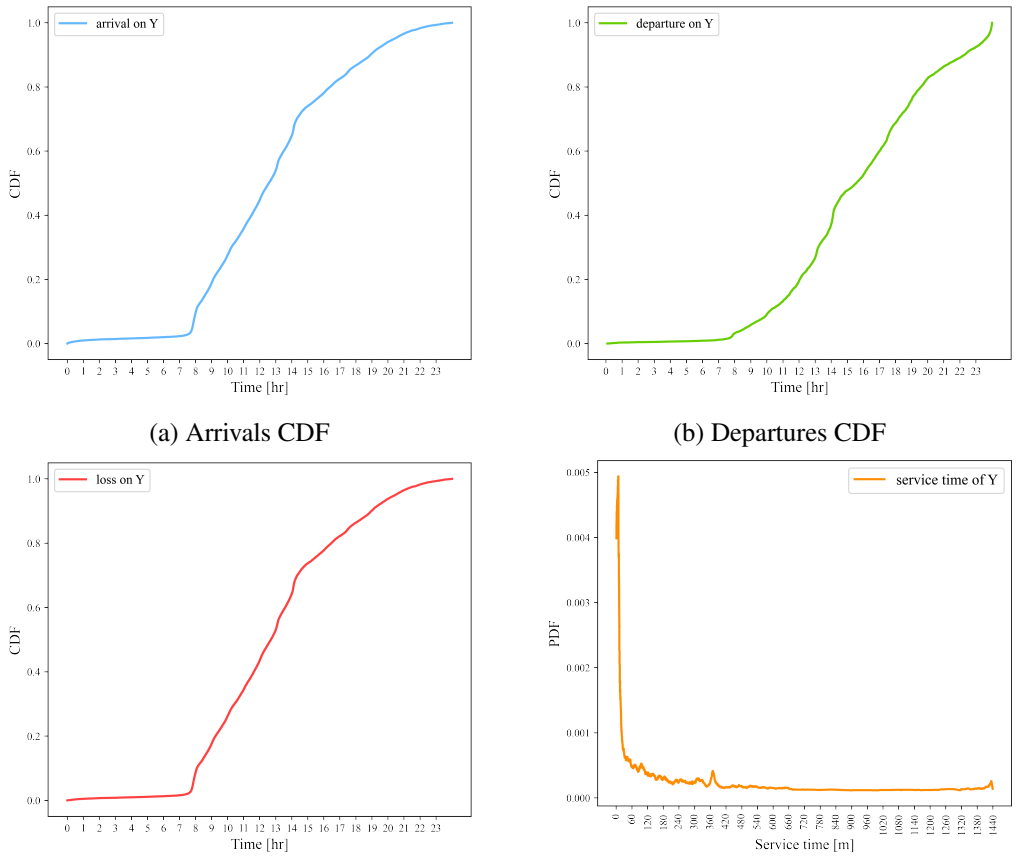


Fig. 4.10 Queue model for scanner-Y

The loss curve in Fig. 4.11 is similar to the arrival curve as expected and explained in the previous case (Sec. 4.2.1). From the service time plot in Fig. 4.11 (d), most of the detected devices have stayed for a shorter period of time. However, a minor peak is noticed after 360 minutes (6 hours). Which means, there are groups staying for 6 hours near the scanner-*Y*. Again, here, further examination is necessary to justify the moment, thus let's call the phenomenon P_2 for now.

In order to verify phenomenons P_1 and P_2, we can look at the hour-by details from the PDF of the arrival and departures of the model in Fig. 4.12. Unlike the first case of scanner-*X* (Sec. 4.2.1), the major peaks are on the 8th hour for the arrival and on the 14th hour for the departure. Thus, P_1 matches the peak value of the departure distribution, while the absolute time difference between the peak arrival and the peak departure complements the phenomenon P_2. Unlike the *Politecnico di Torino* groups



(a) Arrivals CDF

(b) Departures CDF

(c) Losses CDF

(d) Service time PDF

Fig. 4.11 Distribution of events on scanner-Y

in which the daily events take place every 90 minutes, this group, with phenomena P_1 and P_2, has daily routines running for 6 hours. It should be noted that this scanner (scanner–Y), which is located at the gate of *Politecnico di Torino*, is very close to the gate of the *Liceo Scientifico Galileo Ferraris* high school. Hence, the scanner has captured the schedules of the high school student groups, the starting and ending times of their daily activities.

Similarly, the arrival and departure event model is confirmed with a detailed daily presentation of the similar week, as shown in Fig. 4.13 and Fig. 4.14. In the figures, the detected devices are groped in a ten-minute window time frame. As expected, the timing of groups, especially those of larger size, harmonizes with the arrival and departure PDF peaks.

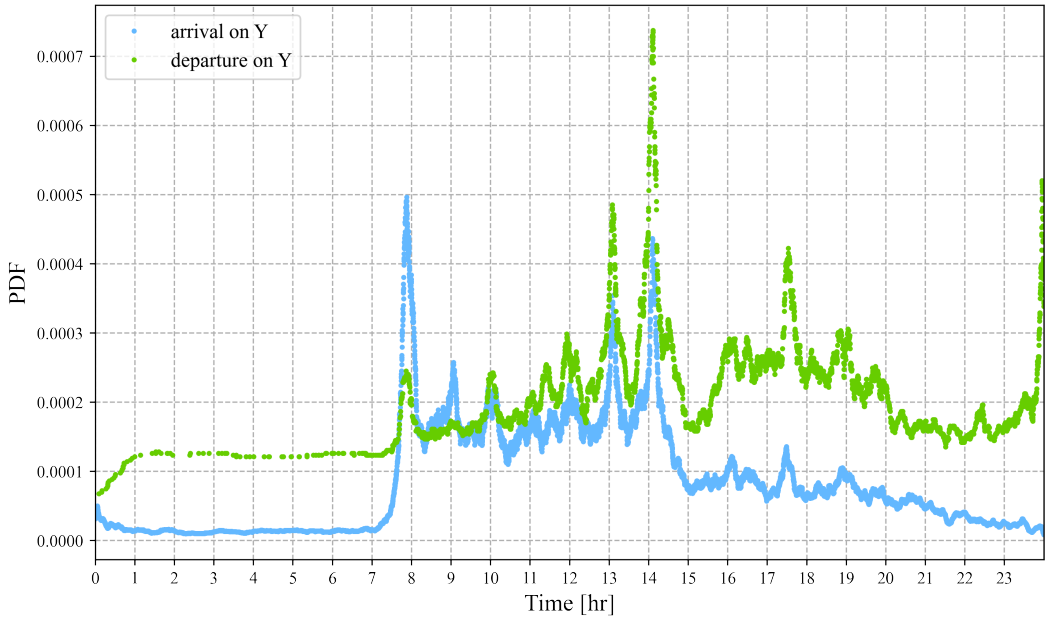


Fig. 4.12 Arrival and departure PDF for scanner-Y

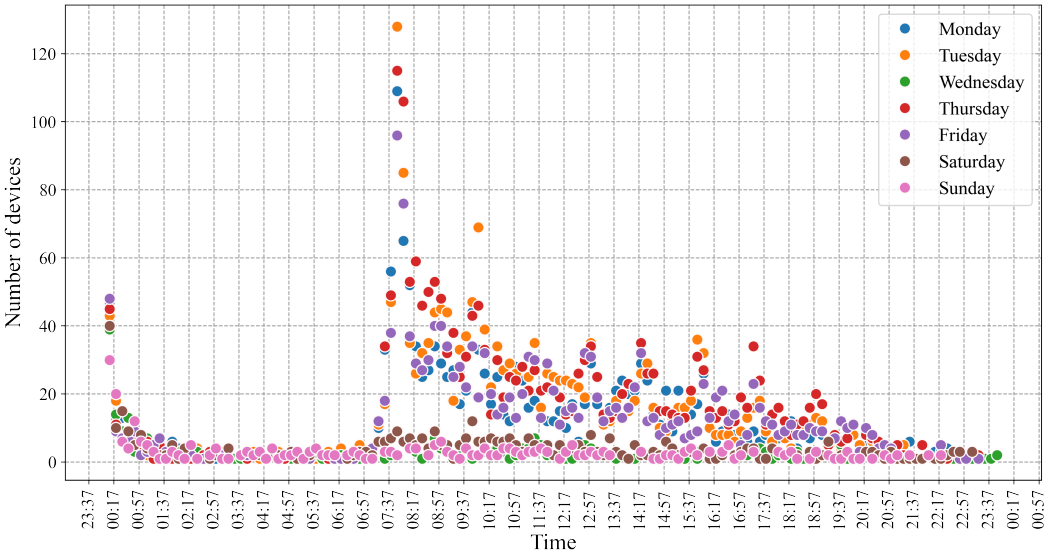


Fig. 4.13 Arrivals on scanner Y

4.2.3 Scanner-$X\&Y$ events

In the last case, it considers both the scanners together instead of just one separately. On this model, the arrival event, the first probe detection, can be either by scanner-X or by scanner-Y. Similarly to the previous cases, the device can be silent or out of

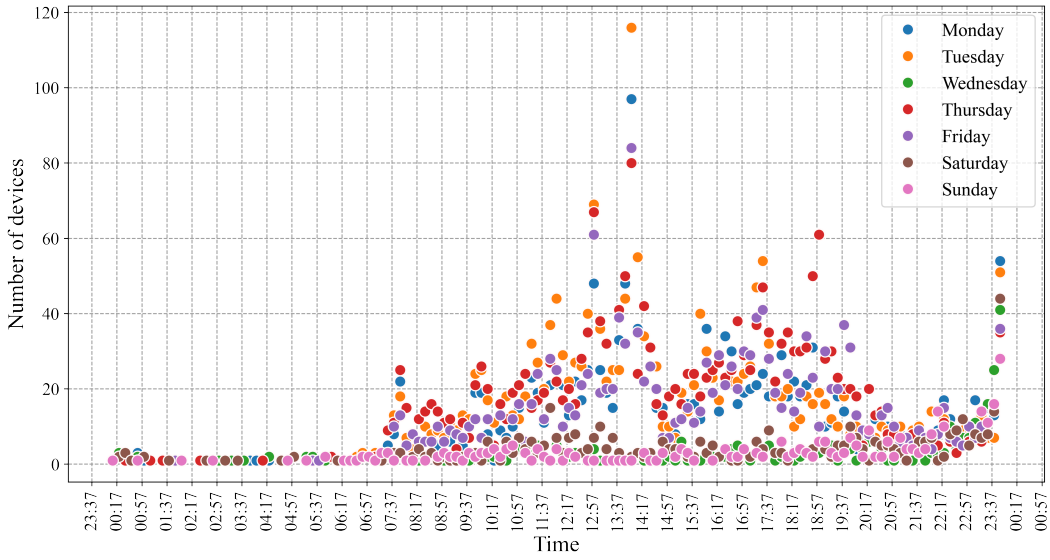


Fig. 4.14 Departures on scanner Y

reach for the entire day after transmitting a single probe, where the event is marked as a loss in the system. However, if the device keeps transmitting the probe at least once after its first probe over the day, and if detected by at least one of the scanners, its last probe of the day will be labeled for the departure event. Thus, the absolute time difference between the arrival and the departure event is the service time as per the model. During the service time, probes from the device can jump between the scanners before departure, for example, first detected by scanner-X, then by scanner-Y, then again by scanner-$X(X \rightarrow Y \rightarrow X)$, or first detected by scanner-Y, then by scanner-X, then again by scanner-$Y(Y \rightarrow X \rightarrow Y)$.

When the two scanners are combined, the most important phenomena captured by the separate models are not hidden in the integrated model. As we can see from the arrival and departure graph from Fig. 4.16, for instance, phenomenon P_1 of the second case (Sec. 4.2.2) is detected at the same 14^{th} hour as in the current model. The assumptions and the results remain similar for the loss curves between the first two models and the last model, evidently as well, as seen in Fig. 4.16.

In Fig. 4.16 (d), the service time of the third model summarizes both the first and second models. Besides the short service times detected by the scanner-X and scanner-Y, the minor peak of the 6 hour service time from the second case (Sec. 4.2.2) is per-

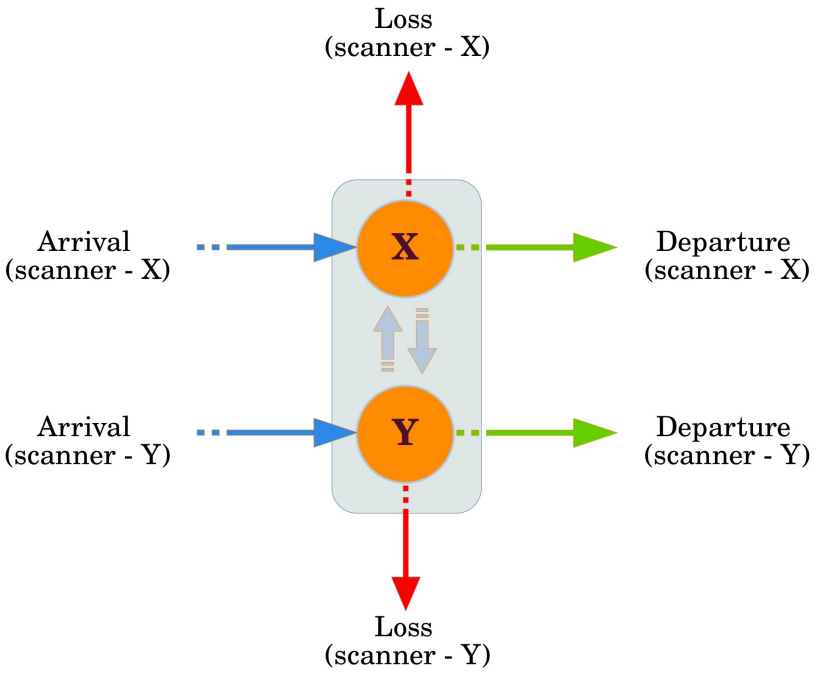
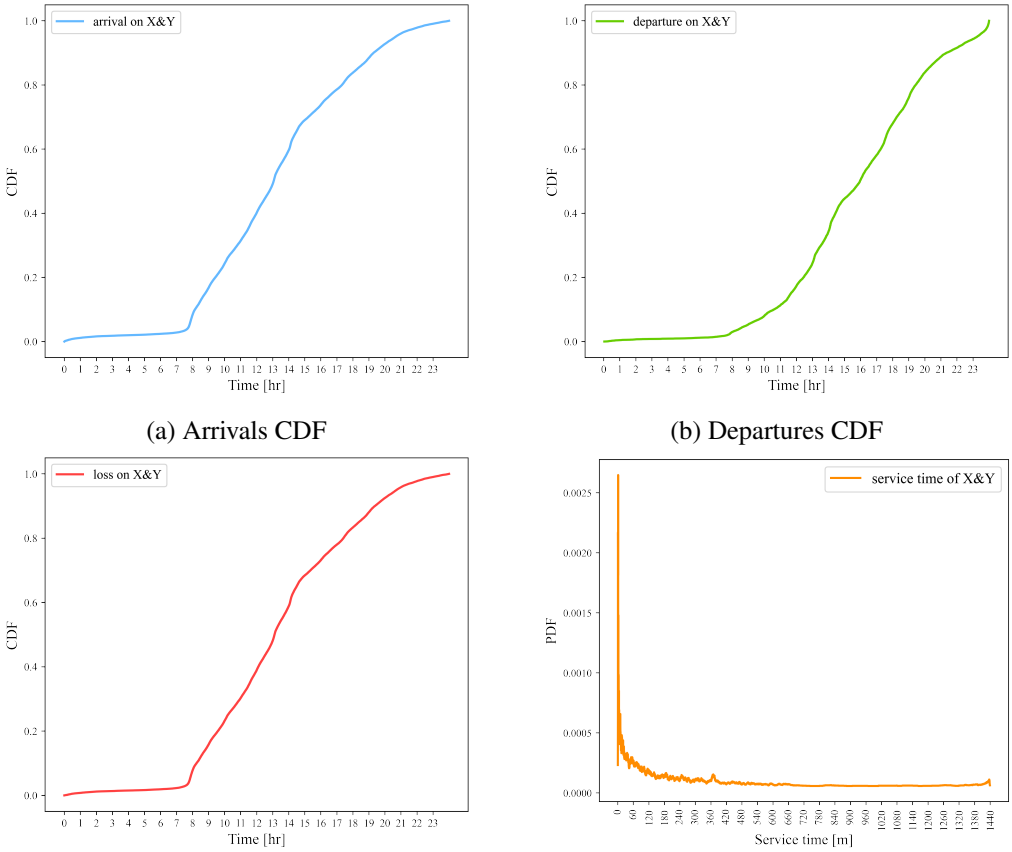


Fig. 4.15 Queue model for scanner-X&Y

ceived in the result of the current model.

The final PDF plot of arrivals and departures on Fig. 4.17 recaps the hour-by-hour main events seen by the first two separate cases. The pivotal events, such as the 90 minute effect for the campus groups and the 6-hour daily high school lessons, are spotted from the distribution of the third model.



(a) Arrivals CDF

(b) Departures CDF

(c) Losses CDF

(d) Service time PDF

Fig. 4.16 Distribution of events on scanner-X&Y

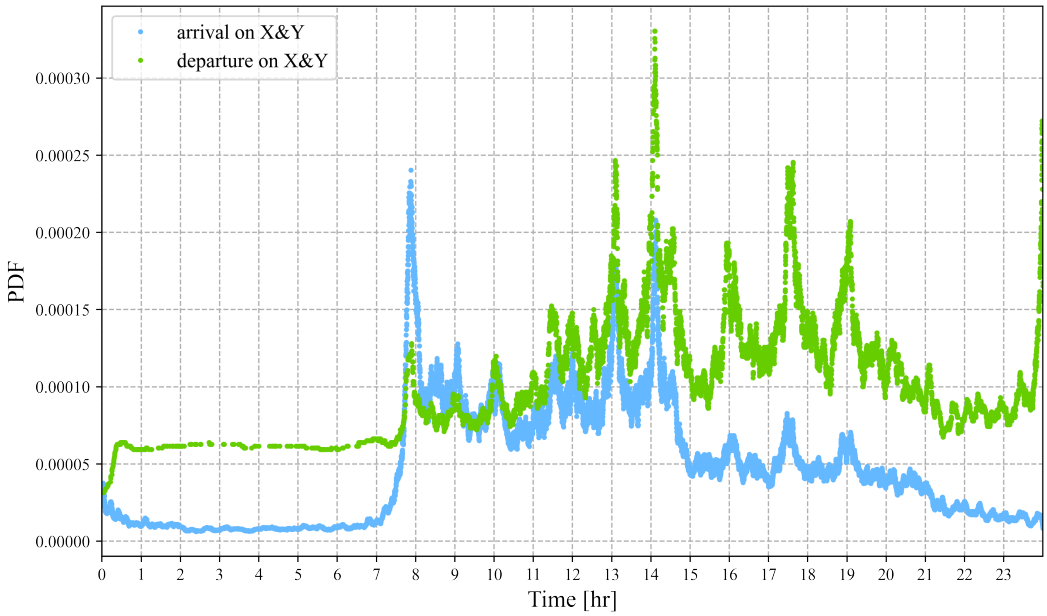


Fig. 4.17 Arrival and departure PDF for scanner-X&Y

4.3 Final remark

In this chapter, we have looked at statistical analysis of groups from WiFi probe request frames and MAC addresses incorporated with the frame. From the statistics, we have detected events. The analysis was supported by a model, where the distributions of the model were able to complement the analyzed statistics.

Chapter 5

MEC-based Extended Virtual Sensing for Mobility Safety Service

Part of the work presented in this chapter has been published in [37, 38]:

- *Avino, Giuseppe and Bande, Paolo and Frangoudis, Pantelis A. and Vitale, Christian and Casetti, Claudio and Chiasserini, Carla Fabiana and Gebru, Kalkidan and Ksentini, Adlen and Zennaro, Giuliana, "A MEC-Based Extended Virtual Sensing for Automotive Services," in IEEE Transactions on Network and Service Management, vol. 16, no. 4, pp. 1450-1463, Dec. 2019, doi: 10.1109/TNSM.2019.2931878.*
- *Avino, Giuseppe and Giordanino, Marina and Franzoudis, Pantelis A. and Vitale, Christian and Casetti, Claudio and Chiasserini, Carla Fabiana and Gebru, Kalkidan and Ksentini, Adlen and Stojanovic, Aleksandra, "A MEC-based Extended Virtual Sensing for Automotive Services," 2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), 2019, pp. 1-6, doi: 10.23919/EETA.2019.8804512.*

Road traffic injuries are causing consequential damage to societies. According to the World Health Organization (WHO), the leading cause of death for children and young adults aged less than 30 is road traffic injuries [39]. In order to address this, the automotive sector is focused on traffic safety applications. The goal of this work is to deal with the road traffic injuries that could occur during mobility. For this reason, the Extended Virtual Sensing (EVS) system was developed in order to

predict future accidents involving vehicles and pedestrians. The system is capable of determining collisions at road intersections, thus making early notifications for the concerned entities to avoid accidents. The system collects mobility data from vehicles and pedestrians to have a broad environmental view before the required warnings. Since such safety systems relay on low latency infrastructure, the EVS system is based on MEC technology. In order to implement the MEC system, we have used the *OpenAirInterface* (OAI) standard platform. Furthermore, an urban environment with intersecting roads is modeled to test the EVS system, where on the model the traffic flows of vehicles and pedestrians are emulated. The performance of the implemented safety system is also included in the chapter.

5.1 MEC Architecture

Multi-access Edge Computing (MEC) enables MEC applications to run at the network edge, within an operator network. These MEC applications, comprising important functions, are virtualized applications supported by a virtualization framework such as Virtual Machines and container applications inside a MEC host. MEC services are delivered through the MEC applications. Therefore, the MEC platform enables a setting where applications can discover, advertise, consume, and offer services.

The MEC host containing the MEC platform and applications is responsible for providing the necessary resources in order to facilitate the required computations and storage along with the networking services. Each MEC application specifies rules and requirements, such as the maximum latency, corresponding to the service. These demanded specifications must be confirmed by the MEC system level management, which is in charge of an overall view of the MEC system.

The management entity handles the services and the available resources through MEC Orchestrator. Besides validating rules and requirements, the orchestrator controls the on-boarding and offloading as well as audits the integrity and authenticity of applications. Furthermore, the MEC Orchestrator is responsible for relocating applications when needed. Requests for instantiation and termination processes of the applications are via the CFS portal and from device applications. Before the

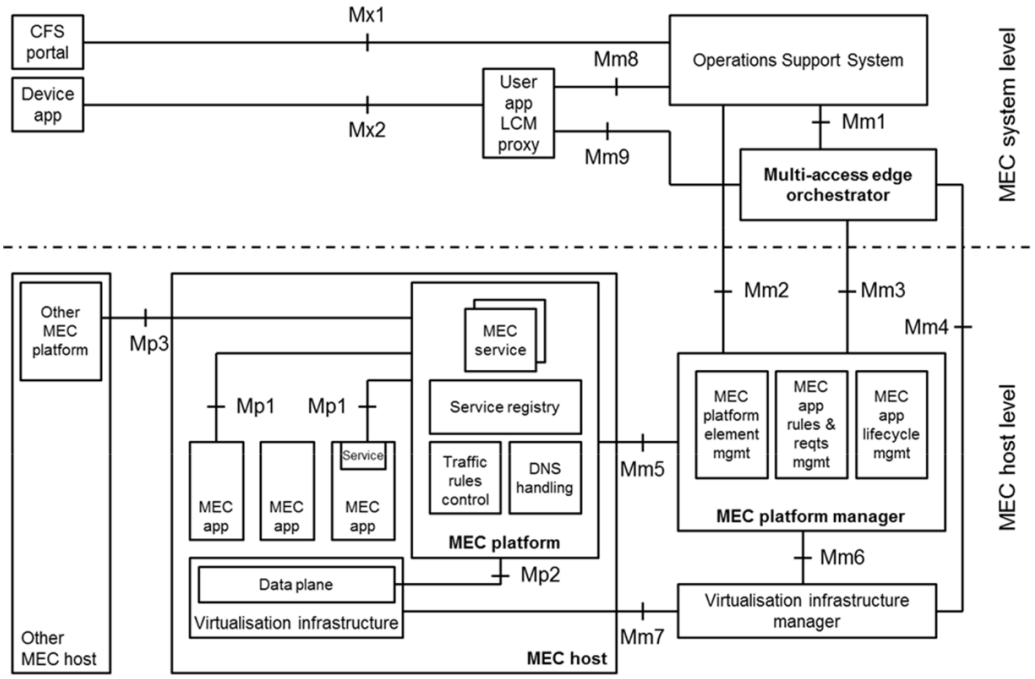


Fig. 5.1 Multi-access edge system reference architecture (ETSI, [2])

Orchestrator further processes and manages the applications, the Operations Support System, which is another component of the management entity, is accountable for deciding whether to grant or not the forwarded requests.

5.2 Extended Virtual Sensing Application

Throughout mobility, vehicles and pedestrians can get enhanced safety services from various information sources. The vehicles in particular can take advantage of the embedded ADAS sensor systems (i.e., radar, camera, etc.) to acquire environmental data, unlike the pedestrians. However, the Extended Virtual Sensing system is capable of delivering momentous services for both vehicles and pedestrians. The central server of the EVS system collects data from vehicles and pedestrians' smartphones concerning their whereabouts and situations, thus developing safety-related information from the collected corporative data in order to provide environmental awareness to users. The EVS, apart from the ADAS system, can be considered as a *virtual sensor* for the cars, while pedestrians benefit from an auxiliary safety-related geographical sense. Hence, based on the notification message from the server, the

users or the autonomous system can take measures in order to avoid safety risks.

The EVS system has three main benefits for users. The first is larger coverage area support, i.e., the EVS can manage a wider area than the ADAS sensors. Secondly, vehicles that are not equipped with a full-fledged set of ADAS sensors as well as pedestrians can be assisted through the *virtual sensor*. Most importantly, the third benefit is the prediction service on future safety threats during mobility.

In this work, the particular service of the EVS is compatible with one of the 5GT project safety use cases, which provides road safety service for automotive. The goal of the service is to spot and avoid forthcoming collisions between vehicles as well as with pedestrians. Since these types of collisions mostly occur at road intersections, the *future collision sensor*, referred to as a *collision detector*, focuses on such crossroads. In order to perform accurately, the collision detector relies on two procedures.

The first one is the message exchange process in which vehicles and pedestrians periodically transmit mobility status information towards the EVS central server, i.e., the collision detection application. Thus, from the collected messages, the server is responsible for identifying vehicles and pedestrians that are at risk and notifying them prior to the imminent collision. The message exchange is explained further in the next section (Sec. 5.2.1).

The second most important part of the safety service on the EVS system is the collision detection algorithm, which computes the prospective collisions for the vehicles and pedestrians. The details of the execution of the algorithm are explained in Section (Sec. 5.2.2). Thus, in the following sections, the message exchange process and the collision detection process are presented.

5.2.1 EVS messages exchange

The collision detector system on the EVS is based on two types of messages. The first is transmitted from the vehicles and pedestrians towards the collision detection server, while the second is directed from the server towards vehicles and pedestri-

ans at risk. These messages, used in road safety services, are defined in the ETSI reference architecture document. The first messages generated by the vehicles and pedestrians are called Cooperative Awareness Messages (CAM), and the second messages forwarded by the server are called Decentralized Environment Notification Messages (DENM), as defined by the standard.

The CAMs sent by the users should mainly contain the current position, speed, and heading information at least. However, additional information such as the type and size of the vehicle could be encoded in the message. According to the ETSI, these beacon packets should be small in size, specifically 50 to 300 bytes, and should be transmitted at least once every one second, unless there is a sudden change in position, speed, or heading information, in which case CAMs must be generated every 100 milliseconds for three consecutively at least. The 100 milliseconds frequency is triggered by the speed change of 5 meters per second, position change of 4 meters, and heading change of 4 degrees or more.

The second type of message, DENMs, is transmitted not frequently but when necessary, unlike the CAMs. It is only necessary to transmit the DENMs when the system detects a forthcoming impact between two entities. Thus, the notification message is forwarded only to the concerning entities continuously, every 100 milliseconds, until there is a response from the entity or the safety threat is clear.

5.2.2 Collision detection

The heart of the EVS system for road safety is the collision detection application. The goal of this application is to prevent future mobility accidents. It, therefore, leverages the cooperative messages collected from the vehicles and pedestrians. The CAMs are the foundation of the collision detection algorithm in order to compute the future encounters for the users. The algorithm, as seen in Alg. 1, is called and executed per CAM since with every CAM there is new information. As the new cooperative awareness message arrives from an entity, it is matched with all the latest CAM of the remaining entities in the geographical area. As a result, if two or more entities are at risk of colliding in the future, each entity is tagged for the DENM

notification message. The collision detection algorithm is examined more in detail in the following section.

Collision Detection algorithm

The CD algorithm is critical to the service because it is the only unit that can predict future events. It can notify entities at risk of collision ahead if the situation is inevitable. In order to do the operation, the CD algorithm takes as an input the latest position, velocity, and acceleration of the entity along with the remaining set of latest CAMs in the environment (Line 0). After collecting the inputs, an empty set, as a temporary memory, is created to record entities at risk (Line 1). Then, the future trajectory of the entities is computed from the input (Line 2 and Line 3). Next, the future trajectory is computed for all the remaining entities in the area (Line 6 and

Algorithm 1 Collision detection pseudocode

Require: $\vec{p^0}, \vec{v}, \vec{a}, \beta$
1: $\zeta \leftarrow \emptyset$
2: $p_x(t) \leftarrow p_x^0 + v_x t + \frac{1}{2} a_x t^2$
3: $p_y(t) \leftarrow p_y^0 + v_y t + \frac{1}{2} a_y t^2$
4: **for all** $b \in \beta$ **do**
5: **read** $\vec{\tilde{p}^0}, \vec{\tilde{v}}, \vec{\tilde{a}}$ from b
6: $\tilde{p}_x(t) \leftarrow \tilde{p}_x^0 + \tilde{v}_x t + \frac{1}{2} \tilde{a}_x t^2$
7: $\tilde{p}_y(t) \leftarrow \tilde{p}_y^0 + \hat{v}_y t + \frac{1}{2} \tilde{a}_y t^2$
8: $D(t) \leftarrow (p_x(t) - \tilde{p}_x(t))^2 + (p_y(t) - \tilde{p}_y(t))^2$
$= \left[p_x^0 - \tilde{p}_x^0 + (v_x - \tilde{v}_x) t + \frac{1}{2} (a_x - \tilde{a}_x) t^2\right]^2 +$
$\left[p_y^0 - \tilde{p}_y^0 + (v_y - \tilde{v}_y) t + \frac{1}{2} (a_y - \tilde{a}_y) t^2\right]^2$
9: $\tau \leftarrow t : \frac{\mathrm{d}}{\mathrm{d}t} D(t) = 0$
10: **for all** $t^\star \in \tau$ **do**
11: **if** $t^\star < 0$ **or** $t^\star > t2c_t$ **then**
12: **continue**
13: **end if**
14: $d^\star \leftarrow \sqrt{D(t^\star)}$
15: **if** $d^\star \leq s2c_t$ **then**
16: $\zeta \leftarrow \zeta \cup \{b\}$
17: **break**
18: **end if**
19: **end for**
20: **end for**
21: **return** ζ

Line 7), after taking the inputs from the CAM database (Line 5). Following that, the trajectories of the entities are matched with each other in order to compute the distance between them (Line 8). Then, the time instance when the two entities were at their closest distance from each other is extracted in Line 9. Since the equation is biquadratic, more than one time instance is returned on Line 9. Thus, in Line 10, all the instances are taken into consideration.

In Line 11, two conditions are checked: whether the returned time is negative or greater than a time threshold value (the time threshold is explained in Sec. 5.2.2). A negative t on the result indicates a growing distance between the two entities, while a value greater than the time threshold indicates an early, if not unnecessary, warning since situations may be altered (i.e., entities can break, change heading, etc. with a larger time window). Thus, if the entities are moving further apart from each other or the time is early, the collision threat is ignored for the moment (Line 12). However, if the time instance t is positive and below the time threshold, the distance between the entities at time t is extracted (Line 14) to be matched with another threshold value (Line 9), a distance threshold (explained in Sec. 5.2.2). If the extracted distance is less than the distance threshold value (Line 15), the entities are expected to collide, hence the entities are pushed into the set ζ for notification (Line 16). Finally, the set ζ is forwarded to the DENM transmitter as explained in Sec. 5.2.1.

Collision detection parameters

In the algorithm, there are two important parameters that can affect the performance of the EVS system; a time threshold (T_2C) and a distance threshold (S_2C) parameter. While the first is referred to in the work as time-to-collision, the second is called space-to-collision.

The first parameter, T_2C, is the time difference between the time when the two entities are close to each other and the current running time of the algorithm. The threshold value considers the time needed for transmitting the warning (DENM), the reaction time after receiving the DENM, the breaking to stop the entity, and a safe margin. Therefore, if this threshold is lower, collision could be inevitable and detection will be considered *late*. Instead, when this threshold is higher, false positive warnings

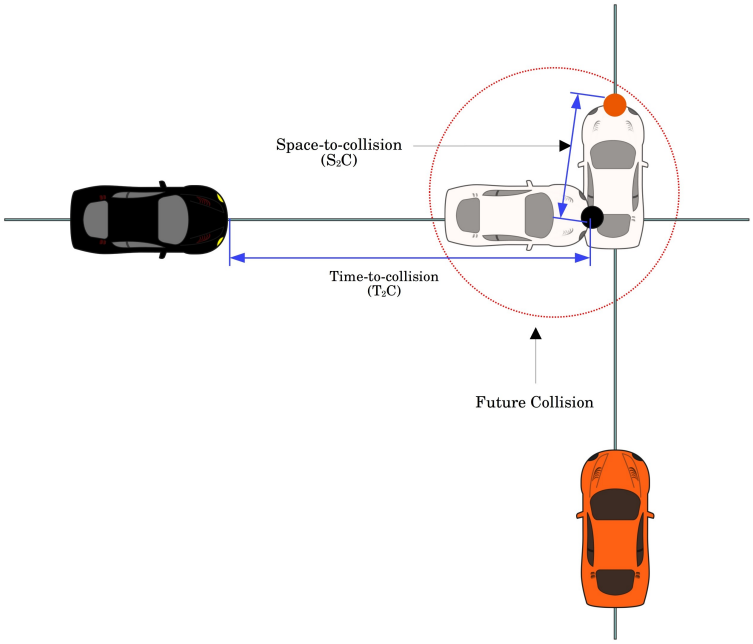


Fig. 5.2 Collision detection parameters: S_2C and T_2C

could be transmitted since one of the entities could at least change the parameters in milliseconds, i.e., make breaks, change headings, etc. Hence, the value must balance the required time in order to improve the performance of the algorithm.

The second parameter, S_2C, is the minimum distance between the two entities over the trajectory made by the algorithm. The threshold value controls the performance of the system. When this threshold is lower, the forthcoming collisions could be undetected by the algorithm, so true negative effects can take place. On the other hand, when the S_2C value is higher, the entities could receive a false positive alert since there may not be a collision if the entities are further apart from each other. In the algorithm, this value should also consider the size of the entities to cover the whole body on the trajectory of the likely collision ahead. Thus, for larger entities, the value should increase in order to detect collisions at the tail of the entity as well.

5.3 Testbed design and implementation

In order to achieve our goal, a testbed has been implemented for the MEC-based EVS system. The testbed includes three classes: users, a cellular infrastructure, and the EVS service, as shown in Fig. 5.3.

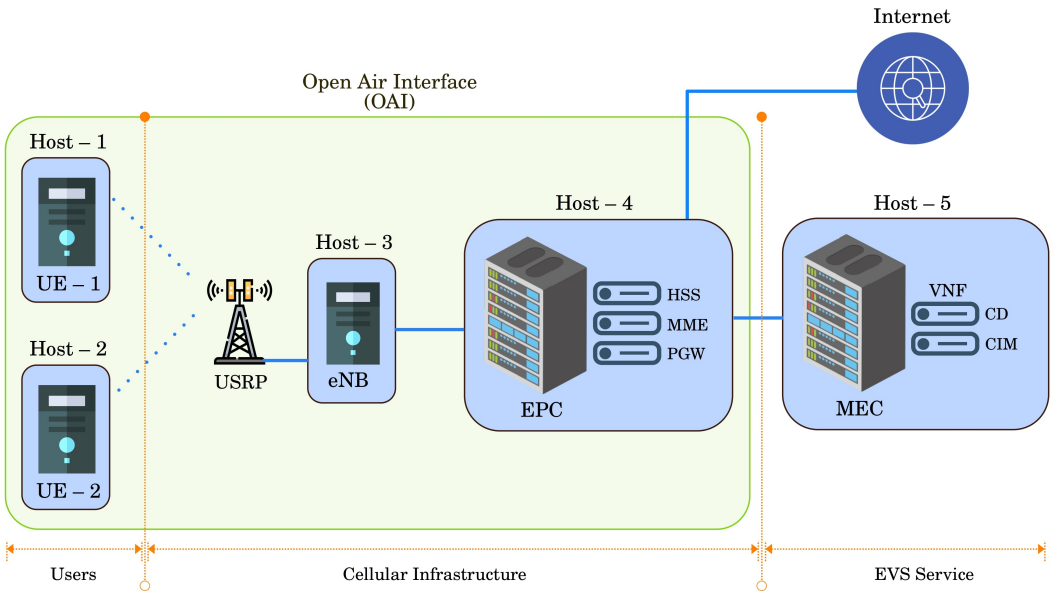


Fig. 5.3 MEC-based Extended Virtual Sensing testbed

The first class, the users, has two physical hosts whose main responsibility is to transmit predetermined CAMs of vehicles and pedestrians as programmed. These messages are real packets communicated over a cellular network, where the hosts act as User Equipment (UE) of the LTE infrastructure. In order to generate the cooperative awareness messages from the UE, the traffic flow of vehicles and pedestrians is emulated on the SUMO traffic simulator, and traces of each flow are recorded in advance. In the simulation, position, speed, acceleration, and heading information are encoded and logged every 100 milliseconds for each entity. Thus, the two UE hosts can emulate the traffic flow of vehicles and pedestrians from the traces and forward every single CAM on time towards the EVS service provider through the cellular infrastructure.

The UEs and the LTE cellular infrastructure are implemented based on the OAI standard. The over-the-air communication between the UEs and the eNB is supported by the USRP radio board on the testbed, as shown in Fig. 5.3. On the platform, the three main systems, the HSS, the MME, and the PGW, are virtualized and incorporated into the EPC host. Hence, each and every CAM from the UEs can reach the MEC server through the PGW.

The MEC host, located at the edge of the testbed, has two VNFs in order to deliver the EVS service to the users. The first VNF is the Cooperative Information Manager (CIM), whose role is to receive the CAM messages from the UEs and manage them. Thus, the CIM features two modules: a receiver module and an information management module (IM). The receiver module is accountable for three tasks: collecting the CAMs from the UEs over the UDP; decoding the CAMs to extract the information; and forwarding the information towards the IM that is in charge of storing the decoded information. The second VNF of the EVS system located on the MEC host is the Collision Detector (CD). This virtualized instance includes three modules: the CD manager, the CD algorithm, and the DENM transmitter. The CD manager requests the set of CAMs from the IM of the CIM. During this procedure, CAMs only from the monitored area can be extracted. After that, the CD manager forwards the latest CAMs to the CD algorithm. Thus, the CD algorithm computes the trajectories of all the vehicles and pedestrians from the monitored area in order to detect collisions ahead, thus tagging entities at risk. Once vehicles and pedestrians are tagged for future collision by the CD algorithm, information about the tagged entities is forwarded to the third module of the CD VNF, the DENM transmitter. Therefore, the DENM transmitter is responsible for notifying the entities at risk through the DENM protocol.

5.4 Proof of concept scenario

In support of the work, we have modeled mobility in an urban environment. The main purpose of the EVS service is to enhance safety by avoiding imminent collisions during mobility. The modeled environment is designed in a way to induce collisions, which the EVS system can counter. For this reason, there are two intersections for the impacts between vehicles and three pedestrian crossings for the misfortune of

pedestrians on the road map (shown in Fig. 5.4).

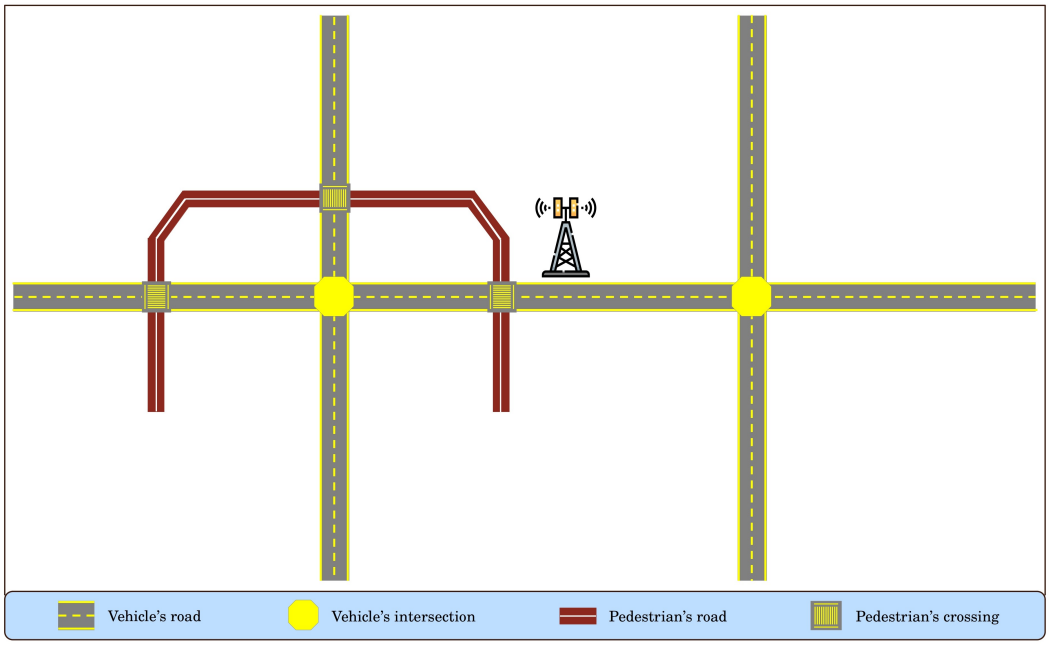


Fig. 5.4 Simulated mobility scenario

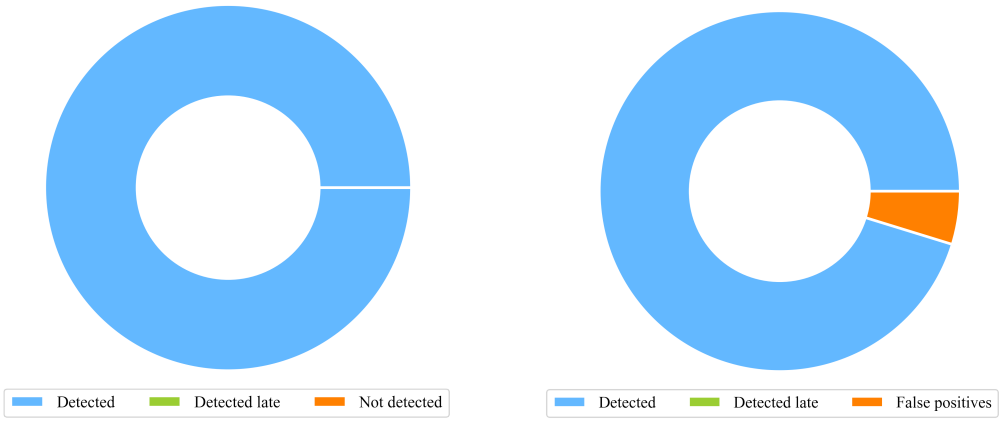
On top of this road map, the traffic flow of vehicles and pedestrians is modeled with the Poisson process of rate of $\lambda_v = 0.7$ and $\lambda_p = 0.15$ independently, where λ_v is the arrival rate for vehicles and λ_p is the arrival rate for pedestrians. The chosen average arrival rates provide a stable system but fabricate collisions at the intersections and the crossings.

The scenario, which is carried out by the SUMO traffic simulator, has extra parameters such as the dimension and speed of the entities. Thus, on the simulator, the dimensions are $4.3m \times 1.8m$ and $0.2m \times 0.4m$ respectively, for the vehicles and pedestrians. As for the speed, the maximum is 50kph (13.9m/s) for vehicles and 7.2kph (2m/s) for pedestrians.

5.4.1 Collision Detection performance

From the discussion made on Sec. 5.2.2, the performance of the algorithm, thus the EVS system, depends on the S_2C and T_2C threshold values. While the T_2C determines the detection to be *late* or on time, the S_2C controls the warnings to be *false positive* when too small or *undetected* (i.e., true negatives) when too large. Based on that, the thresholds are set empirically on the system for the chosen scenario (Sec. 5.4), as follows.

- $V_2V : S_2C = 3.7$ meters, $T_2C = 2.5$ seconds, where V_2V refers collision between vehicles
- $V_2P : S_2C = 1.4$ meters, $T_2C = 0.9$ seconds, where V_2P refers collision between vehicle and pedestrian



(a) Collision detection (b) False positives

Fig. 5.5 Collision between vehicles

Note that the separate values between V_2V and V_2P are due to the different reaction behaviors of vehicles and pedestrians as well as the different sizes of the entities.

The results in Fig. 5.5 and Fig. 5.6 show the performance of the algorithm. With the selected parameters, all the inevitable collisions are detected. However, false positive warnings are noticed as well. When details of the false negatives are investigated, Fig. 5.7 shows the warnings are caused by short-distance errors of less than one meter.

(a) Collision detection (b) False positives

Fig. 5.6 Collision between vehicles and pedestrians

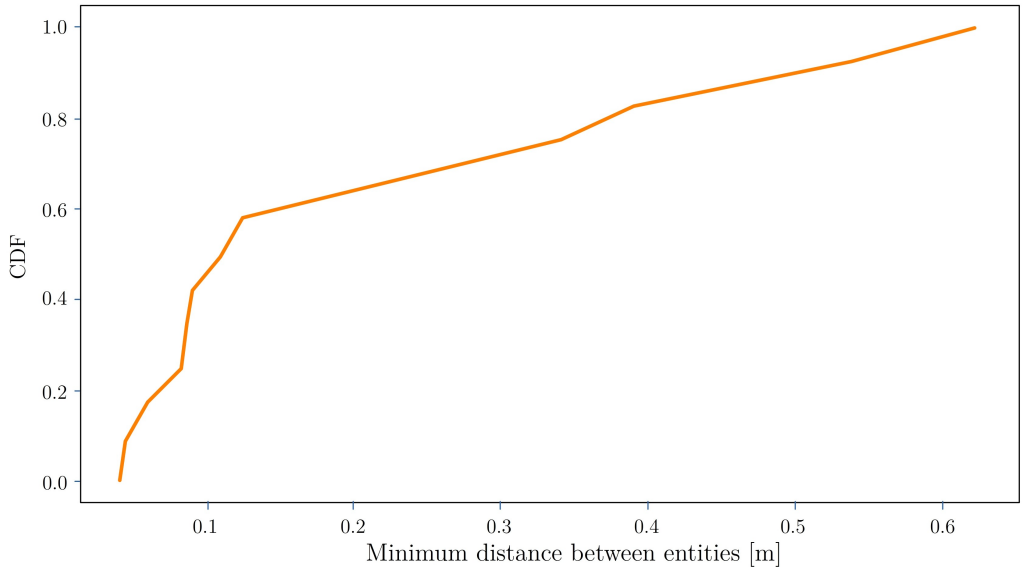


Fig. 5.7 CDF of false positives

The reason for having these false positives is the behavior of the traffic simulator. The SUMO simulator is a discrete-time simulator, thus in the simulation the stepping time is set to 100 milliseconds, i.e., any value lower than this will drag the system down. Instead, the trajectory computation made by the algorithm is continuous. Hence, while the algorithm detects the collision perfectly, the discrete time simulator skips the events between 100 milliseconds. It should be noted that, according to the simulation, a vehicle can move 1.3 meters over 100 milliseconds and a pedestrian 0.2

meters over the skipped duration. Therefore, the false positive results are expected to be the effects of the simulator and the algorithm.

5.5 Final remark

The chapter has covered a MEC-based EVS system, a mobility safety service based on a "virtual sensor" system. The system was able to detect collisions ahead between vehicles as well as between vehicles and pedestrians. Mobility traffic was modeled in an urban environment to test the performance of the system, and the result was a success.

Chapter 6

Conclusions

With respect to people's mobility, we have made contributions in order to provide support for safety-related and other services in general. Systems that can facilitate the support of environmental services as well as safety services have an immense impact on users during mobility. In regard to that, we have offered what we believe could assist the interested sectors through this work: mobility tracking and safety support systems. These systems basically collect mobility data from people while moving in order to deliver the required services. Hence, we have used two types of sensor systems for the purpose of monitoring mobility in a given geographical area.

The first one is an IoT based WiFi sensor, while the second one is a virtual sensor system. The first sensor system, the IoT-based WiFi sensor, is used for detecting WiFi signals from mobile devices without the need for third-party applications for the purpose of tracking mobility. In order to achieve this, we have leveraged two types of WiFi sensors; off-the-shelf commercial WiFi sensors and ad-hoc designed WiFi sensors. With the help of these sensors, WiFi probe request frames in particular, were collected during the mobility of people. These probe request packets contain information such as the MAC address of the device, which is used by the mobility tracking system. To protect people's privacy when collecting these data, which are considered personal by the EU GDPR, we have implemented a privacy-preserving scheme. Moreover, we have used the 5G-EVE architecture along with the scanners as a testbed for the mobility tracking application. On top of our testbed, we have made ground-truth experiments with scenarios of mobility flow. Finally, we have

applied our mobility tracking methodologies in order to make classifications between mobility patterns and flows, thus validating the work.

The second sensor system supports both vehicles and pedestrians with safety services during mobility. The system relays on two types of messages exchanged between users and the server. The first type of message, encoded with the latest mobility data such as position and heading, is transmitted by vehicles and pedestrians. Instead, the second message is a safety warning sent from the EVS system, which monitors the geographical area. In order to do this, the EVS system has included a safety service system, in particular a Collision Detector. The purpose of the CD system is to predict collisions ahead in the environment from the messages sent by vehicles and pedestrians. Thus, upon detecting imminent collisions, the CD will notify the concerned entities before the incident, which is highly critical for avoiding traffic accident injuries. In order to carry out and test the system, we have implemented a testbed based on the OAI standard. Furthermore, we have modeled the traffic flows of vehicles and pedestrians in an urban environment, which allowed us to measure the performance of the system. According to the results, the work was successful.

Appendix A

Published and Submitted Contents

Below, the list of my published papers during the Ph. D. program:

[27] **Kalkidan Gebru**, Marco Rapelli, Riccardo Rusca, Claudio Casetti, Carla Fabiana Chiasserini, Paolo Giaccone, "Edge-based passive crowd monitoring through WiFi Beacons," Computer Communications, Volume 192, 2022, Pages 163-170,ISSN 0140-3664, https://doi.org/10.1016/j.comcom.2022.06.003

[28] **K. Gebru**, *"A Privacy-preserving Scheme for Passive Monitoring of People's Flows through WiFi Beacons,"* 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), 2022, pp. 421-424, doi: 10.1109/CCNC49033.2022.9700591.

[32] **K. Gebru**, C. Casetti, C. F. Chiasserini and P. Giaccone, "IoT-based Mobility Tracking for Smart City Applications," 2020 European Conference on Networks and Communications (EuCNC), 2020, pp. 326-330, doi: 10.1109/EuCNC48522.2020.9200941.

[37] Avino, Giuseppe and Bande, Paolo and Frangoudis, Pantelis A. and Vitale, Christian and Casetti, Claudio and Chiasserini, Carla Fabiana and **Gebru, Kalkidan** and Ksentini, Adlen and Zennaro, Giuliana, "A MEC-Based Extended Virtual Sensing for Automotive Services," in IEEE Transactions on Network and Service Management, vol. 16, no. 4, pp. 1450-1463, Dec. 2019, doi: 10.1109/TNSM.2019.2931878.

[38] Avino, Giuseppe and Giordanino, Marina and Franzoudis, Pantelis A. and Vitale, Christian and Casetti, Claudio and Chiasserini, Carla Fabiana and **Gebru, Kalkidan**

and Ksentini, Adlen and Stojanovic, Aleksandra, "A MEC-based Extended Virtual Sensing for Automotive Services," 2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), 2019, pp. 1-6, doi: 10.23919/EETA.2019.8804512.

List of acronyms

5G-EVE	5G European Validation platform for Extensive trials
5GT	5G-Transformer
ADAS	Advanced Driver-Assistance Systems
AE	Application Entity
ASCII	American Standard Code for Information Interchange
ARM	Advanced RISC Machines
CAM	Cooperative Awareness Messagge
CD	Collision Detection
CDF	Cumulative Distribution Function
CFS	Customer Facing Service
CIM	Cooperative Information Manager
CoAP	Constrained Application Protocol
CSE	Common Service Entity
DENM	Decentralized Environment Notification Message
eNB	Evolved Node B
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
EU	European Union
EVS	Extended Virtual Sensing
GDPR	General Data Protection Rule
GPS	Global Positioning System

HSS	Home Subscriber Server
HTTP	HyperText Transfer Protocol
IE	Information Element
IM	Information Management
iOS	iPhone Operating System
IoT	Internet-of-Things
JSON	JavaScript Object Notation
LTE	Long-Term Evolution
M2M	Machine-to-Machine
MAC	Media Access Control
MEC	Multi-access Edge Computing
MME	Mobility Management Entity
MQTT	Message Queuing Telemetry Transport
NSE	Network Service Entity
OS	Operating System
OAI	OpenAirInterface
PDF	Probability Density Function
PGW	Packet Data Network Gateway
PNL	Preferred Network Lists
RAM	Random Access Memory
RP	Raspberry PI
RSSI	Received Signal Strength Indicator
S_2C	Space-to-Collision
SHA	Secure Hash Algorithm
SSID	Service Set Identifier
SUMO	Simulation of Urban Mobility (open source software)
T_2C	Time-to-Collision
TIM	Telecom Italia

UDP	User Datagram Protocol
UE	User Equipment
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
V_2P	Vehicle-to-Pedestrian
V_2V	Vehicle-to-Vehicle
VNF	Virtual Network Function

Bibliography

[1] "Probe Request Frame." https://www.rfwireless-world.com/Terminology/WLAN-probe-request-and-response-frame.html.

[2] M. Etsi, "Multi-access edge computing (mec); framework and reference architecture," *ETSI GS MEC*, vol. 3, p. V2, 2019.

[3] K. I. Shah, M. Khan, S. Abbas, Z. Hasan, and A. Fatima, "Intelligent transportation system (its) for smart-cities using mamdani fuzzy inference system," *International Journal of Advanced Computer Science and Applications*, vol. 9, 01 2018.

[4] J. Freudiger, "How talkative is your mobile device? an experimental study of Wi-Fi probe requests," in *ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pp. 1–6, 2015.

[5] A. E. C. Redondi and M. Cesana, "Building up knowledge through passive wifi probes," *Computer Communications*, vol. 117, pp. 1–12, 2018.

[6] J. Weppner, B. Bischke, and P. Lukowicz, "Monitoring crowd condition in public spaces by tracking mobile consumer devices with WiFi interface," in *ACM UbiComp*, pp. 1363–1371, 2016.

[7] F. Wang, X. Zhu, and J. Miao, "Semantic trajectories-based social relationships discovery using WiFi monitors," *Personal and Ubiquitous Computing*, vol. 21, no. 1, pp. 85–96, 2017.

[8] E. Kalogianni, R. Sileryte, M. Lam, K. Zhou, M. Van der Ham, S. Van der Spek, and E. Verbree, "Passive WiFi monitoring of the rhythm of the campus," in *AGILE International Conference on Geographic Information Science*, pp. 1–4, 2015.

[9] A. Di Luzio, A. Mei, and J. Stefa, "Mind your probes: De-anonymization of large crowds through smartphone WiFi probe requests," in *IEEE INFOCOM*, pp. 1–9, 2016.

[10] C. Chilipirea, C. Dobre, M. Baratchi, and M. van Steen, "Identifying movements in noisy crowd analytics data," in *2018 19th IEEE International Conference on Mobile Data Management (MDM)*, pp. 161–166, 2018.

[11] A. Basalamah, "Crowd mobility analysis using WiFi sniffers," *IGI IJCSSA*, vol. 7, no. 12, pp. 374–378, 2016.

[12] U. G. Acer, G. Vanderhulst, A. Masshadi, A. Boran, C. Forlivesi, P. M. Scholl, and F. Kawsar, "Capturing personal and crowd behavior with Wi-Fi analytics," in *International Workshop on Physical Analytics*, pp. 43–48, 2016.

[13] A. J. Fernández-Ares, A. M. Mora-Garcia, M. I. García-Arenas, P. García-Sánchez, G. Romero, S. M. Odeh, and P. A. Castillo, "A novel wireless mobility monitoring and tracking system: Applications for smart traffic," *IGI IJCSSA*, vol. 4, no. 2, pp. 55–71, 2016.

[14] Y.-E. Sun, H. Huang, W. Yang, S. Chen, and Y. Du, "Toward differential privacy for traffic measurement in vehicular cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 4078–4087, 2022.

[15] B. Soundararaj, J. Cheshire, and P. Longley, "Estimating real-time high-street footfall from Wi-Fi probe requests," *International Journal of Geographical Information Science*, vol. 34, no. 2, pp. 325–343, 2020.

[16] M. Uras, R. Cossu, and L. Atzori, "Pma: a solution for people mobility monitoring and analysis based on wifi probes," in *2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)*, pp. 1–6, 2019.

[17] G. Pipelidis, N. Tsiamitros, M. Kessner, and C. Prehofer, "HuMAn: Human movement analytics via WiFi probes," in *IEEE PERCOM*, 2019.

[18] P. Reichl, B. Oh, R. Ravitharan, and M. Stafford, "Using WiFi technologies to count passengers in real-time around rail infrastructure," in *IEEE ICIRT*, pp. 1–5, 2018.

[19] L. Zhu, H. Tong, L. Lou, and Y. Xiong, "A passenger flow monitoring method in Hongqiao hub area based on gridded Wi-Fi sniffing," in *IEEE IMCEC*, pp. 1052–1057, IEEE, 2018.

[20] L. Gallo and J. Haerri, "Unsupervised long-term evolution device-to-device: A case study for safety-critical v2x communications," *IEEE vehicular technology magazine*, vol. 12, no. 2, pp. 69–77, 2017.

[21] Z. Riaz, D. Edwards, and A. Thorpe, "Sightsafety: A hybrid information and communication technology system for reducing vehicle/pedestrian collisions," *Automation in construction*, vol. 15, no. 6, pp. 719–728, 2006.

[22] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt, "Wreckwatch: Automatic traffic accident detection and notification with smartphones," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 285–303, 2011.

[23] M. R. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, "Cooperative collision avoidance at intersections: Algorithms and experiments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1162–1175, 2013.

[24] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "Lte for vehicular networking: a survey," *IEEE communications magazine*, vol. 51, no. 5, pp. 148–157, 2013.

[25] Z. Xu, X. Li, X. Zhao, M. H. Zhang, and Z. Wang, "Dsrc versus 4g-lte for connected vehicle applications: A study on field experiments of vehicular communication performance," *Journal of advanced transportation*, vol. 2017, 2017.

[26] Z. Hameed Mir and F. Filali, "Lte and ieee 802.11p for vehicular networking: a performance evaluation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, pp. 1–15, 2014.

[27] K. Gebru, M. Rapelli, R. Rusca, C. Casetti, C. F. Chiasserini, and P. Giaccone, "Edge-based passive crowd monitoring through wifi beacons," *Computer Communications*, 2022.

[28] K. Gebru, "A privacy-preserving scheme for passive monitoring of people's flows through wifi beacons," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 421–424, IEEE, 2022.

[29] C. Badii, P. Bellini, A. Difino, and P. Nesi, "Sii-mobility: An iot/ioe architecture to enhance smart city mobility and transportation services," *Sensors*, vol. 19, no. 1, 2019.

[30] "GDPR." https://www.gdpr.net/.

[31] M. Nitti, F. Pinna, L. Pintor, V. Pilloni, and B. Barabino, "iabacus: A wi-fi-based automatic bus passenger counting system," *Energies*, vol. 13, no. 6, p. 1446, 2020.

[32] K. Gebru, C. Casetti, C. F. Chiasserini, and P. Giaccone, "Iot-based mobility tracking for smart city applications," in *2020 European Conference on Networks and Communications (EuCNC)*, pp. 326–330, IEEE, 2020.

[33] "European 5G validation platform for extensive trials." https://www.5g-eve.eu/.

[34] G. Pipelidis, N. Tsiamitros, M. Kessner, and C. Prehofer, "Human: Human movement analytics via wifi probes," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 370–372, IEEE, 2019.

[35] L. Zhu, H. Tong, L. Lou, and Y. Xiong, "A passenger flow monitoring method in hongqiao hub area based on gridded wi-fi sniffing," in *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 1052–1057, IEEE, 2018.

[36] J. Weppner, B. Bischke, and P. Lukowicz, "Monitoring crowd condition in public spaces by tracking mobile consumer devices with wifi interface," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pp. 1363–1371, 2016.

[37] G. Avino, P. Bande, P. A. Frangoudis, C. Vitale, C. Casetti, C. F. Chiasserini, K. Gebru, A. Ksentini, and G. Zennaro, "A mec-based extended virtual sensing for automotive services," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1450–1463, 2019.

[38] G. Avino, M. Giordanino, P. A. Franzoudis, C. Vitale, C. Casetti, C. F. Chiasserini, K. Gebru, A. Ksentini, and A. Stojanovic, "A mec-based extended virtual sensing for automotive services," in *2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*, pp. 1–6, 2019.

[39] "Road traffic injuries." https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.