POLITECNICO DI TORINO
Repository ISTITUZIONALE

IoT for Real Time Presence Sensing on the 5G EVE Infrastructure

*Original*
IoT for Real Time Presence Sensing on the 5G EVE Infrastructure / Rusca, Riccardo; Casetti, CLAUDIO ETTORE; Giaccone, Paolo. - ELETTRONICO. - (2021). (Intervento presentato al convegno Mediterranean Communication and Computer Networking Conference (MedComNet 2021) tenutosi a Online nel 15-17 June 2021) [10.1109/MedComNet52149.2021.9501245].

*Availability:*
This version is available at: 11583/2897852 since: 2021-09-06T08:18:39Z

*Publisher:*
IEEE

*Published*
DOI:10.1109/MedComNet52149.2021.9501245

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*
IEEE postprint/Author's Accepted Manuscript

©2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

20 April 2024

# IoT for Real Time Presence Sensing on the 5G EVE Infrastructure

Riccardo Rusca, Claudio Casetti, Paolo Giaccone
Politecnico di Torino and CNIT, Italy

***Abstract*—One of the most widely advertised capabilities of 5G targets the massive Machine-Type Communication (mMTC) giving the development of Internet of Things (IoT) solutions center stage in the new generation of mobile networks. In this paper, we address the possibility of detecting people on city streets thanks to deployment of commercial sensors, connected to the 5G network, that capture WiFi probes transmitted by people's smart devices. We first outline the motivation of such a scenario. Then, we illustrate our implemented architecture and present the results detected in an area near the Politecnico di Torino within the 5G EVE H2020 project. We show that our architecture can monitor real-time data coming from the installed sensors and thus estimate the number of people present in an area by simply collecting anonymized MAC addresses and timestamps from smart devices of passers-by.**

## I. Introduction

Internet-of-Things (IoT) systems have started to permeate people's everyday lives, both at home and in Smart City environments. With the coming of age of 5G networks, several use cases have been developed to improve the citizens' quality of life and to allow city administrations to introduce new services. Indeed, one of the definitions of a "Smart City" relates to an urban area that uses different types of IoT electronic sensors to collect data and efficiently provide, through the analysis of the captured data, resources and services to improve life in the city. The Smart City concept integrates information and communication technologies (ICT) and various physical devices connected to the city's IoT network; this permits to calibrate and optimize the efficiency of the operations and services that the city offers to its citizens, and it also makes it easier to monitor what is happening in the city in real-time and how the city is evolving. But there are even more compelling reasons: following the pandemic caused by COVID-19, the detection of gatherings of people as well as the counting of people in transit areas has become crucial in order to monitor and ensure the safety of the communities. The role of IoT devices for the purpose of detecting people gatherings has thus become crucial.

In this paper, we address the problems of people counting and of detecting the main mobility patterns, presenting an architecture that uses data collected by commercial WiFi scanners. These sensors are capable of capturing probes messages passively transmitted by smartphones or other smart devices that are trying to identify known nearby WiFi Access Points (APs) or other devices to connect to. Of course, the limitations of this approach are that, in principle, we cannot know if the people are either on foot or on a vehicle. And, when detected by the scanners, if they carry with them just one smart device or more than one.

The rest of the paper is organized as follows: Sec. II presents the implemented architecture, the IoT scenario, and provides some details on which data are collected and their structure, while the retrieval and filtering processes are analyzed in-depth in Sec. III. Some results about the main detected mobility patterns and the implemented real-time dashboard are then provided in Sec. IV. A discussion of related work is in Sec. V and we draw our conclusions in Sec. VI.

## II. 5G EVE Scenario and Testbed Architecture

The 5G EVE architecture [1] supporting our IoT use case includes an edge cloud environment, where edge applications are realized by combining multiple Virtual Network Functions (VNFs). In Figure 1, we can see the details of the testbed specifically implemented for the mobility tracking use case. The WiFi scanners operate on the available channels in the 2.4 and 5 GHz ISM bands. All the scanners are connected through the RAN (*Radio Access Network*) to the OneM2M server [2], an open architecture for the provision of IoT services.

All the data stored in the OneM2M platform are retrieved by the MOB ("MOBility tracking") VNF and saved in a local database for subsequent uses. Finally, the experimenter can connect to the VIS ("VISualization") VNF to see in real-time the data collected by the scanners through a web-based visualization tool.

The platform's edge cloud (grey part in Figure 1) runs in a dedicated data center available within Politecnico di

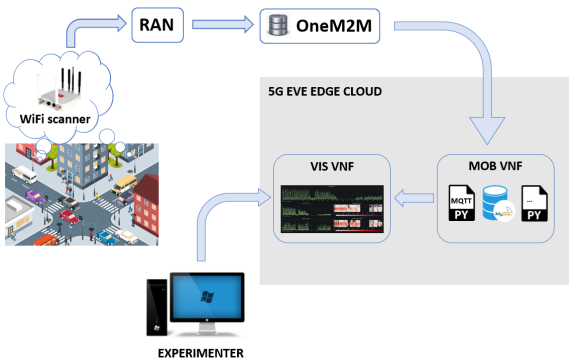



Fig. 1: 5G EVE Edge Cloud testbed architecture by which the data collected by the WiFi scanner is processed through two VNFs [3]


Fig. 2: Coverage map of the 5G EVE testbed with the 6 Wi-Fi scanners, in the area between the Politecnico di Torino campus and the Porta Susa train station.

Torino. Thanks to the use of the edge paradigm, the proposed architecture can scale horizontally with the number of scanners installed. Furthermore, the use of two VNFs to realize the entire mobility tracking application provides high flexibility in terms of required hardware resources.

The experimental testbed covers the area between the main campus of Politecnico to Porta Susa, which is the major transit train station of Torino. The area is characterized by a large inflow and outflow of people that every day move into and out of the campus. In this area, next to the avenue, a bike path is also present, so the WiFi scanners can capture the devices carried by people using different transportation means (e.g., foot, bike, electric scooter, car, motorbike). Six Meshlium Libelium WiFi scanners [4] were installed as shown in Figure 2: two of them were installed inside the campus close to entry gates, and the others were installed on top of traffic lights, with the support of the municipality and utilities companies. The scanners were configured to passively capture the "Probe Requests" that are periodically sent by mobile devices while searching for known WiFi networks. Probe requests messages include a unique device identifier (i.e., the MAC address) and some device capabilities (e.g., supported 802.11 standards). They can be directed to one specific Access Point (AP) by indicating its SSID (*Service Set IDentifier*) or broadcasted to all APs within range.

Each scanner is connected to the 5G EVE platform [1] through a cellular connection. According to factory settings, every 51 seconds the scanners group the information about all the detected devices during the last sampling period and every two minutes they upload the collected data to the OneM2M platform [2].

Each sample collected and uploaded by each scanner is described by the following fields:

- *the MAC address of the detected device*. This field is anonymized by digesting the device MAC address through an SHA-224 function directly on the WiFi scanner. The default hash function available in the Libelium scanners digests the MAC address together with the current time, not allowing the identification of the same MAC address at different times. To avoid this problem, the hashing mechanism was modified to digest just the MAC address without timestamp;
- *the timestamp, with a one-second precision*. Since the sampling occurs every 51 seconds, all the probe requests observed during the same sampling period are recorded with the same sampling period, making it impossible to get a detailed timing sequence of the probe requests. Notably, multiple probe requests coming from the same device during the same sampling period are collapsed into a single sample;
- *the RSSI at the receiver*. This field has not been considered since the Libelium documentation does not explain how the value of RSSI is evaluated (e.g., it is unclear if the value detected is the average or the maximum) and how the RSSI of multiple probe requests received during the same sampling period is computed. Furthermore, as well known in the literature, the RSSI cannot be considered a reliable metric for mobility tracking;
- *the interface vendor*. This field has not been considered since it does not permit to identify the interface uniquely and in most of the cases is equal to "Unknown".

Figure 3 exemplifies a sample in JSON format, reporting the fields described above.

### A. MAC randomization

Some operating systems have devices randomize the MAC address to ensure user privacy, i.e., the device can use a fake random address to send probe requests. This is especially a problem for our monitoring platform because a single device

```
"from": "2020-08-10 00:55:00",
"to": "2020-08-10 00:57:00",
"data[0].RSSI": "-85",
"data[0].Vendor": "Ubiquiti Networks, Inc.",
"data[0].TimeStamp": "2020-08-10 00:55:14",
"data[0].MAC": "464F368FA0A7A0F7992FBE19B9F596AE138CD5DF0B0A1CAE57B6F958",
"data[1].RSSI": "-85",
"data[1].Vendor": "ZyXEL Communications Corporation",
"data[1].TimeStamp": "2020-08-10 00:55:14",
"data[1].MAC": "8C0E3AC26FD518FC1CCEC6D0E1CD9ACE0CDAAC66AE758C1D7BF2C846",
```


Fig. 3: Example of samples for two detected devices in JSON format

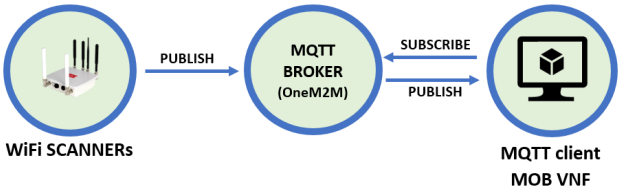



Fig. 4: MQTT publish-subscribe protocol in the MOB VNF

can appear with multiple MAC addresses. Since we count the unique MAC addresses to infer the number of different devices in an area, MAC randomization leads to overestimating the actual number. Both Android and Apple iOS are known to have been using MAC randomization techniques [5], [6]. Recent studies have introduced solutions to overcome this limitation for counting procedures by implementing a sort of de-randomizer [7], an algorithm that is based on the analysis of common characteristics present in different probe requests, which remain constant even with randomized MAC addresses, and other parameters that allow calculating the probability that two random MAC addresses belong to the same device. In our case this de-randomizer is not applicable because for privacy reason all the MAC addresses are anonymized with an irreversible SHA-224 hash function.

## III. Edge-based mobility tracking application

As shown in Figure 1, the mobility tracking application has been implemented by combining two VNFs within the edge cloud. The data available from the OneM2M system is retrieved and processed by the MOB VNF, running on a dedicated linux Virtual Machine (VM). The results are shown in real-time by the VIS VNF, running on another VM and exposing a standard web interface to the experimenters.

### A. The MOBility tracking VNF

The MOB VNF retrieves the data from the remote platform through an MQTT (Message Queuing Telemetry Transport) client. Notably, MQTT is a very lightweight, open transport protocol, based on the publish-subscribe paradigm. As a reminder, the protocol runs over TCP/IP, so it is reliable and it prevents the out-of-order delivery of data.

As depicted in Figure 4, the MQTT client makes a subscription for each topic for which it wants to be notified when new data with these specific topics are uploaded. The mechanism involves creating a Subscription resource, triggering a notification message from the MQTT broker whenever a resource (containing data) is uploaded. When a new message arrives at the broker from the WiFi scanner, it is saved in the OneM2M server, and then it is also sent to the MQTT client where the message is parsed, analyzed and then stored into a local MySQL database.

Before saving the data on the database, two operations occur in sequence: (i) address digesting, (ii) stationary device removal. In the address digesting phase, each anonymized MAC address is represented on 224 bits, corresponding to 56 hexadecimal digits. In order to optimize the implementation in terms of performance and storage, we stored just a subset of bits for each MAC address. This procedure increases the probability that two distinct MAC addresses are stored with the same identifier and thus collide in our database. In order to avoid collisions, we exploited some theoretical results regarding the well-known birthday paradox. The probability $p$ of collision for MAC addresses based on $k$ bits can be estimated as follows:

$$p \approx 1 - e^{-\frac{m^2}{2^{k+1}}}$$

where $m$ is the number of stored MAC address. In our scenario, with almost 15 million MACs corresponding to 10 months of data collected, $p$ is $6 \cdot 10^{-6}$ and for 30 millions MAC $p$ is $2 \cdot 10^{-5}$, i.e., the collision probability is completely negligible.

During the operations for the stationary device removal, MAC addresses of fixed devices around the scanners, like Smart TVs or IoT devices, are detected and inserted in a blacklist to be removed from further processing. These type of devices can be identified thanks to the continuous flows of probe requests observed around the clock, making it very unlikely that they belong to passers-by. Every night, a Python script automatically runs with the aim to find new possible MAC addresses to add to the blacklist, following an empirical procedure, whose pseudocode is shown by Algorithm 1. After retrieval, the MAC addresses are saved into a dictionary $T$, where the MAC $x$ is the key and the value $T(x)$ is an array of timestamps associated with the MAC. Then, for each MAC address the number of timestamps per hour is evaluated. If there were more than 15 timestamps per hour for at least 20 hours out of 24, the corresponding MAC is considered belonging to a stationary device so it is added to the blacklist $\Omega_B$ and no longer considered during the processing and the visualization.

Since October 1st, 2019, we have identified 168 distinct MAC addresses in $\Omega_B$, of which about 40 were detected on the first day, and around 7-8 new ones have been added every month into $\Omega_B$ by the algorithm.

### B. The VISualization VNF

The VIS VNF provides a flexible visualization tool, accessible with a standard web browser, to show the results of the processing occurring in MOB VNF. We chose Grafana [8]

**Algorithm 1** Stationary device detection

```
1:  procedure FIND STATIONARY DEVICES(candidate set Ω of MAC addresses)
2:      let Ω_B = ∅                          ▷ Initialize the set of stationary MAC addresses
3:      for all x ∈ Ω do                     ▷ For each MAC in the candidate set
4:          let T[x] be the set of timestamps associated to x
5:          let counter[24] = []             ▷ Initialize an array with a counter for each hour
6:          for t ∈ T[x] do                  ▷ For each timestamp associated to X
7:              counter[hour(t)]++           ▷ Update the per-hour counter for x
8:          z = 0                            ▷ Counter with the hours with at least 15 observations
9:          for h = 0 → 23 do                ▷ For each hour
10:             if counter[h] ≥ 15 then      ▷ Check if at least observations exist
11:                 z++
12:         if z ≥ 20 then                   ▷ Check if number of hours with many observations
13:             add x to the blacklist Ω_B
14:     return Ω_B                           ▷ Return the set of stationary MAC addresses
```

as a real-time data visualization tool. Grafana is a multi-platform, open-source analytics and interactive visualization web application. It enables the creation of complex monitoring dashboards using interactive query builders on databases and provide interactive charts, graphs and alerts. It is also possible to set threshold values above or below with which to generate alerts.

The two VMs corresponding to MOB and VIS VNFs reside on the same local network, so they communicate with each other by using integrated Grafana API and SQL queries. The MySQL database, containing all the data collected by the WiFi scanners, is the data source for the dashboards.

We implemented different dashboards, one for each scanner and one merging all the scanners' data, as shown in Sec. IV-A. We also created a dynamic website that, for a given time interval, displays a map with the most popular mobility patterns, as shown in Sec. IV-B. This is achieved by Python script running in the VIS VNF.

We also considered alternative solutions with respect to the adopted combination of Grafana with MySQL database. We evaluated two other combinations: Grafana plus Prometheus [9], and Kibana plus Elasticsearch [10], but both have shown some limitations for our use case. Indeed, Prometheus does not allow to save historical series, and data are not accessible for future analysis. Instead, Elasticsearch does not allow to manage the large number of data as collected by our case due to the high overhead when indexing.

## IV. Data analysis

In this section, we describe the main outcome of the large amount of data captured in our testbed. Since October 2019, the total number of detection events has been 56,690,983, corresponding to 24,344,198 distinct MAC addresses. Because of MAC randomization, the latter represents an upper bound on the number of detected devices. In particular, in Sec. IV-A we study the effect caused by COVID-19 restrictions on the number of detection events, in Sec. IV-B we investigate the mobility patterns, and, finally, in Sec. IV-C, we conclude with the analysis of the adopted means of transportation.

### A. Effect of COVID restrictions on the presence of devices

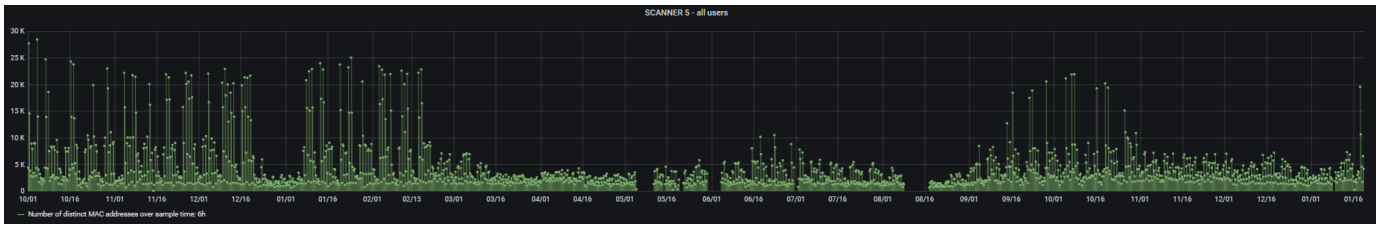
Figures 5 and 6 show a screenshot of the dashboards provided by the VIS NFS and related to two scanners for a time period of about 16 months (from October 1st, 2019 to the first half of January 2021), i.e., covering both the pre-COVID era and the COVID era. Scanner 5 is located within the Politecnico campus but it is very close to the main entrance of a large high school. Scanner 6 is instead located next to the avenue leading to the Porta Susa train station.

The dashboards provided by the VIS VNF show two different representations of the same data: the line chart shows the number of distinct MAC addresses detected over a given sampling period; instead, the heatmaps show the occurrence of detection over time during a particular hour (on the y axis). In the heatmap, the intensity of the red color in each box is proportional to the number of distinct MAC addresses that were detected. This number is affected by the randomization process and thus it represents an upper bound on the actual number of WiFi devices and is related to the number of people present in the covered area (not counting those not carrying detectable devices, of course). Nevertheless, thanks to the law of large numbers, we expect that the actual number of people is a fixed fraction $\gamma$ of the number of distinct MAC addresses. Common sense suggests (even though we cannot prove it) that $\gamma < 1$, given that most people typically carry at most one device with the WiFi interface on. If these reasoning stands, the results can be used to accurately compare the relative number of people in different areas monitored by WiFi scanners.

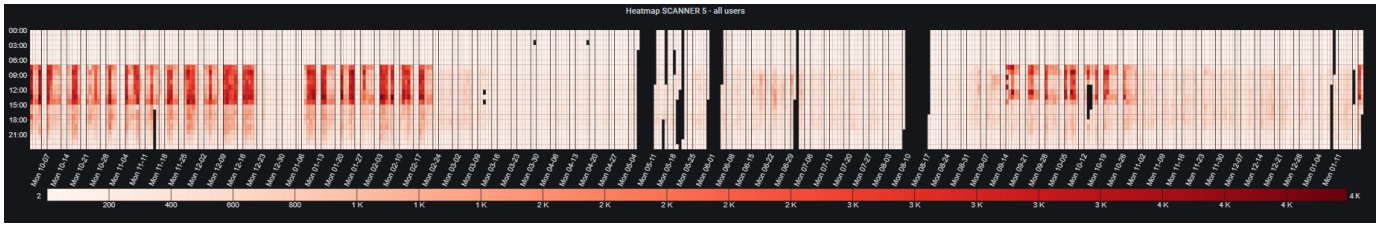
The graphs shown in Figure 5 and Figure 6 present some short periods of missing samples related temporary outages of the scanners due to blackouts, network problems or maintenance activities. Their effect can be considered negligible.

From the line graph of Figure 5 it is possible to observe the effect of the working days, of the holidays and of lockdown periods due to COVID-19, and a number of interesting observations on social behaviors and trends can be highlighted.

The lockdown started in Torino on March 9th, 2020 and the effect on the mobility around the campus and the high school is evident. Indeed, the peaks around 25k MACs detected before March dropped to about 5k detection events from March to September 2020. Also, from the half of September to the first days of November 2020, the school and the university were open again and we can see some peaks of presence. The heatmap of Figure 5 shows very well the periodicity due to the working days in the week. The area is mainly frequented by students either of Politecnico or of the high school. While it is rare to observe people in the area before 7 am, the peaks of presence before COVID are in the time range 7:30 am - 2:30 pm, during working days: they are strongly correlated with the starting time of classes at Politecnico and at the high school, and to the end time of the midday classes at both institutions. Conversely, very few people are around in the evening and early night. Both the line chart and the heatmap of Figure 5 show the effects of the Christmas holiday, highlighted by the lack of students during such a period. After the summer, high-school classes started on September 14th and continued until the end of October, when new COVID-related restrictions interrupted lessons in presence. Instead, the university did not

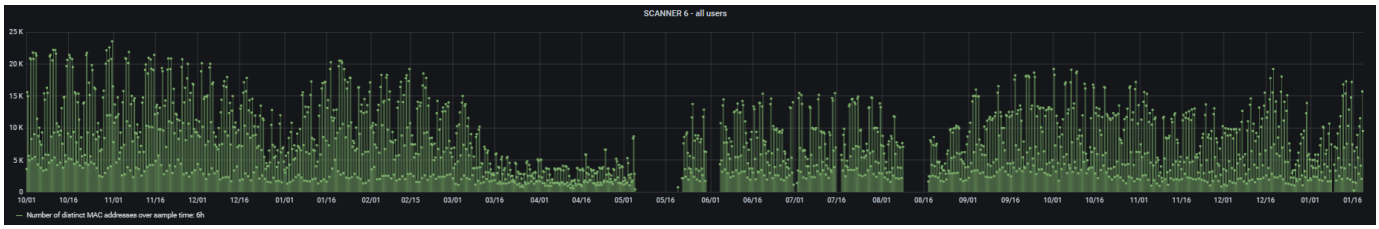



(a) *Line chart scanners 5*

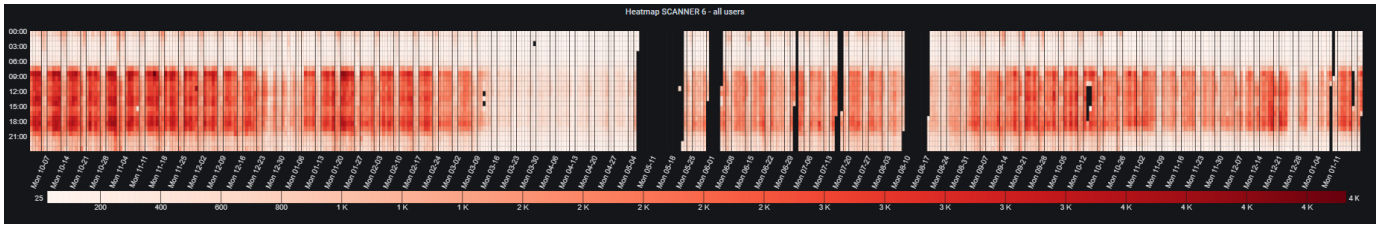



(b) *Heat map scanners 5*

Fig. 5: Dashboard of scanner 5, installed within the Politecnico campus, for the period Oct. 2019-Jan. 2021




(a) *Line chart scanners 6*




(b) *Heat map scanners 6*

Fig. 6: Dashboard of scanner 6, installed near Porta Susa train station, for the period Oct. 2019-Jan. 2021

start classes in presence (except for few hours). Thus, all the peaks in September-October are expected to be due mainly to high-school students. We also expect that the presence of Politecnico employers (faculty and administrative staff) was very limited because of the restrictions to access the offices and of the adopted smart working procedures.

Figure 6 covers a much more crowded area than the one covered by Figure 5, with a broader variety of people due to the closeness to the train station. One very interesting phenomenon that can be seen in the line graph of Figure 6 is how the number of detection events after the lockdown, from May/June, has risen to near the standard values of the months before COVID-19, despite some mobility restrictions. This means that most people transiting in the area are not related to Politecnico, whose classes were still going on in an on-line fashion. This is actually not surprising, due the location far from Politecnico. In the heatmap of Figure 6 we can see how the peak hours reflect the classic working hours, i.e., between 8 am and 7 pm, which is a more extended range with respect students' attendance range observed in Figure 5. The heatmap highlights very well the night life during Saturday evenings before the lockdown in March and the number of people in the early hours of Sunday, from midnight to 2 am, is coherently always higher than all other early hours of the days during the working days.

We now consider specifically the period just before Italy went in lockdown, i.e., the period corresponding to 4 consecutive weeks between the second half of February 2020 and the first half of March 2020. This period of time is interesting since the last two weeks of February corresponded to the (last)

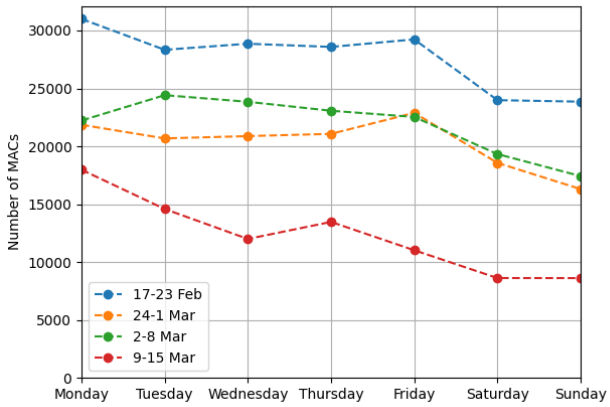



Fig. 7: Number of distinct MAC addresses detected by scanner 9 during February and March 2020, at the time of the initial COVID outbreak in Italy

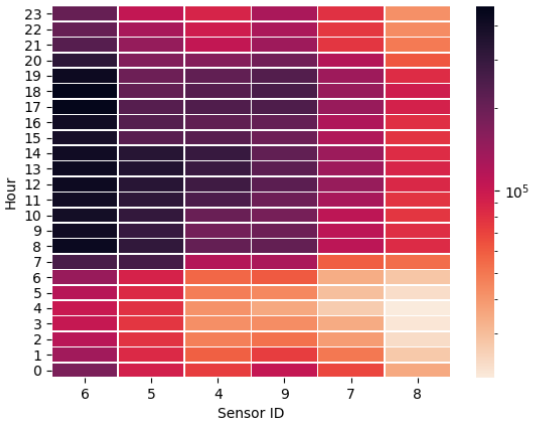



Fig. 8: Per-hour occurrences of single detection for each scanner

in-presence winter session exams at Politecnico di Torino, so the area near the campus was populated by numerous students, professors, and university employees. While from March 9th, 2020, the lockdown officially began throughout Italy, bringing people's movements to the bare minimum.

Figure 7 shows the evolution of the number of devices captured by scanner 9 for each day. We obtained similar results for the other scanners. The number of detection events has dropped showing the effect of COVID restrictions, in particular due to the lockdown during the second week of March. Furthermore, we can see that the number of detection events during the working days for each curve, except for the red one, is almost the same, and then decreases during the weekends due to the lack of students and employees of the Politecnico di Torino, commuting to/from the campus area.

### *B. Analysis of the mobility patterns*

We now investigate the mobility paths and define the *mobility pattern* as the sequence of scanners in which the same MAC address has been detected within a time window of 5 minutes. As an example, the mobility pattern denoted as "498" means that the same MAC address has been detected first by scanner 4 then by scanner 9 and last by scanner 8, and the difference between the timestamps of two consecutive detection events was less than 5 minutes.

Figure 8 shows, through a heatmap in a logarithmic scale, the occurrences of detection for each scanner for different hours of the day. All data used for the charts were captured between October 1st, 2019, and April 25th, 2020. In Figure 8, all the single scanner patterns are represented, i.e., the devices have been detected by a single scanner within a 5 minutes interval of time. This scenario corresponds very likely to two different cases. Either the device leaves the area and no longer passes under another scanners, or the MAC address is changed due to randomization effect. The highest density of devices is observed under scanner 6, and this is expected since the scanner is close to the train station and the street covered by the scanner is the busiest. Moreover, during the hours from 8 to 20, we observed 80% of all the detection events of the day, due to the usual working daily pattern, except for scanner 6, where the time interval extends up to 1 in the morning due to the night life in the area.

Figure 9 shows all observed patterns with multiple scanners. As we can see, the most frequent pattern is the "98", which corresponds to the pedestrian crossing of the main avenue. It is interesting to see how pattern "49" is more frequent than pattern "48"; this can be explained by the fact that many people, on the sidewalk of Politecnico headed north-east towards the train station, prefer to cross the street at the first intersection they find and continue on the opposite sidewalk. Thus, the testbed is also able to capture some micro-mobility patterns, despite the coarse area covered by each scanner.

Figure 10 shows a screenshot of the map obtained by the

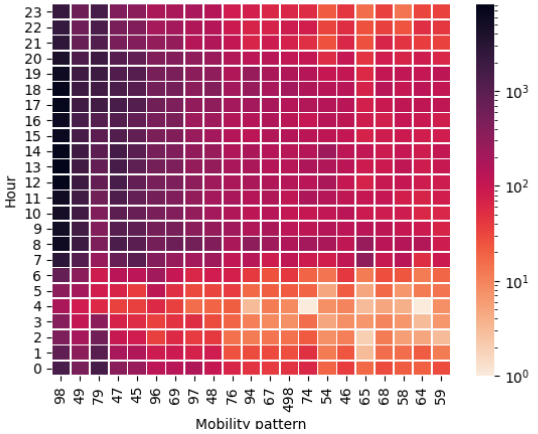



Fig. 9: Per-hour occurrences of mobility patterns captured by multiple scanners

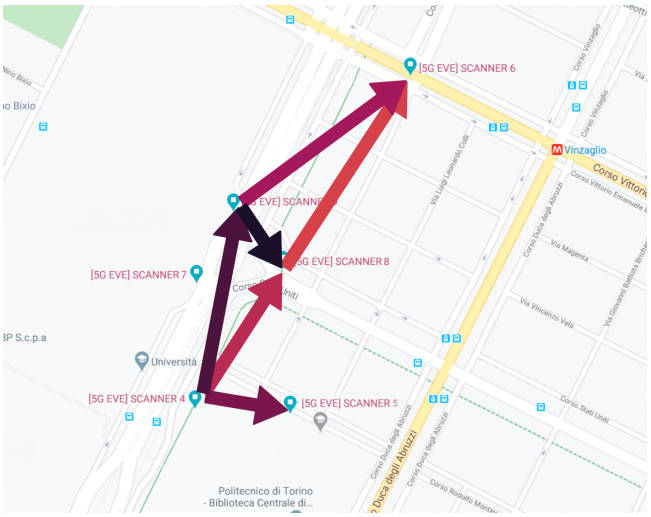



Fig. 10: Mobility paths on February 12th, 2020 between 12:00 and 13:00

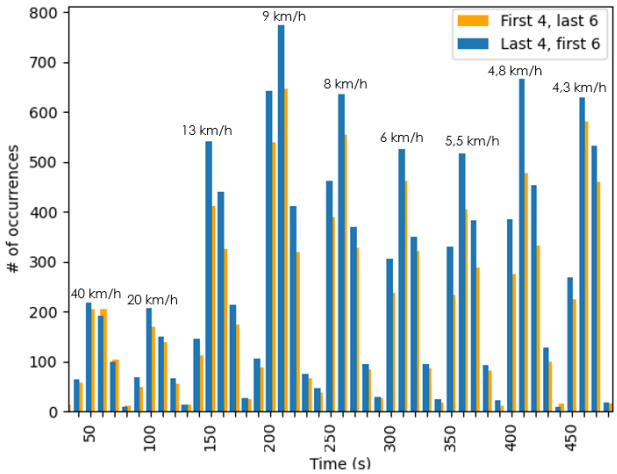



Fig. 11: Distribution of the travel time and estimated speed between scanner 4 and scanner 6 (i.e., mobility pattern "46") from 1/10/2019 to 17/01/2020

web interface exposed by the VIS NFS. Thanks to some colored arrows superimposed on the Politecnico area map, the six most frequent patterns on a given temporal window are highlighted. As in the previous heatmaps, the color tone differentiates the detection density, with darker color corresponding to higher number of detected devices. In detail, Figure 10 shows the mobility path directions of the 12th February 2020 around midday. The most frequent pattern is "98", followed by "49", then "45", "96", "48", and finally "86". So we can expect that the main flow of people was moving from the Politecnico to Porta Susa train station, which is reasonable given that people and students are generally commuting at that hour.

### C. Mobility type

By observing the travel time between different WiFi scanner, we were able to estimate the expected travel speed and the kind of mobility in terms of transportation means. In particular, we show the results just for pattern "46" , because it corresponds to the most distant pair of scanners, with disjoint coverage areas (as shown in Fig. 2) and thus the estimation errors are minimized. Note that such errors depends on both the coarse temporal sampling (we recall that each WiFi scanner produces one sample every 51 seconds) and on the imprecise information about the coverage area, which does not allow to know exactly the physical distance between two samples detected by two distinct WiFi scanners. Indeed, consider the extreme case in which the coverage area of two WiFi scanners are overlapping: a device could be detected at the same time by both scanners, and the travel time would be zero, even if the WiFi scanners are far, resulting in a huge over-estimation of the speed.

The graph in Figure 11 shows the distribution of the travel time between scanner 4 and scanner 6 and the corresponding estimated speed, given that the distance between the two scanners is equal to 550 m. The graph above shows two data series; in particular, the series "last 4, first 6" shown in blue means that the travel time has been calculated taking into account the last detection event on scanner 4 and the first detection event of the same MAC address on scanner 6. Instead, for the series "first 4, last 6", in orange, the calculated time is the difference between the last timestamp detected on scanner 6 and the first of scanner 4. These two series represent respectively the upper bound and the lower bound on the time taken to travel the space between the two scanners. The distribution of the two bounds on the travel times in Figure 11 are almost equivalent, thus the estimated means of transportation is not affected by the uncertainty in the coverage area. This is not surprising, since the two scanners have been chosen to be far from one another.

It is also important to note that cars and motorbikes must cross three traffic light intersections, while people traveling on the cycle path or the sidewalk must only cross one pedestrian crossing. The first peak on the left corresponds to an average travel time of 50 s (i.e., average speed of 40 km/h), while the second about 100 s (i.e., average speed of 20 km/h). Instead, the highest peak corresponds to 210 s (i.e., average speed of 9 km/h). From the travel speeds expressed above, it is evident that the first two peaks on the left, most likely correspond to people moving by cars or motorbikes and the second peak corresponds probably to bikes or electric scooters (often used in the area). Instead, the other three peaks (from 250 s to 370 s) correspond to people on bicycles and some cars stuck in traffic. Finally, all the remaining peaks (above 400 s) are expected to correspond to walking people.

## V. Related Work

Over the years, several technical approaches have been used to address the problem of detecting and counting people in an

urban area, using, e.g., infrared sensors, cameras, pressure sensors, visible light sensors, RFID, UWB, and audio-processing. However, the techniques mentioned above do not provide satisfactory results in relation to the cost of implementation and performance [11].

Some approaches are based on electromagnetic methods that analyze in space and time a signal received which is affected by the presence of the people in the area. All these methods are suitable for small indoor environments and requires specific sources of radio frequency. As an example, the work [12] proposed to count people exploiting the WiFi signal, assuming that the movements of the human body affect the wireless signal reflections, which results in variations in the CSI (Channel State Information). The method works well in a quasi-static indoor scenario, within the same room, as people in a meeting or staffs in an office, but this is very different from our scenario, which is very dynamic, with people moving fast by car, by bicycle or on foot. Another work [13] focused on counting people crossing a doorway using off-the-shelf WiFi devices, by exploiting the reflections of the wireless signal on the human body and by installing special receivers that process the reflected signals. Thus, the approach is tailored to indoor environment and cannot be applied in an outdoor urban scenario with the sensors installed (as in our case) on top of traffic lights.

An alternative approach is to count people using cameras and advanced algorithms for video image processing, but it is quite expensive. For example, the state of the art YOLO_V3 library [14] can identify and count the heads in a 30 FPS video in real-time, but it requires a high-level GPU and a highly equipped server, thus the overall video-based detection platform is too expensive for large-scale use.

Instead, the authors of [15], similarly to our paper, used the probe request messages sent by mobile phones to count the number of passengers in a bus. Through the implementation of filters based on dynamic and static information, they were able to reach an accuracy up to 70%. In this study, however, the randomization of MAC addresses was not considered, which is one of the main problems that undermine the accuracy of people counting, in very dynamic environments.

In [16] we presented some experimental results based on a very preliminary version of the 5G EVE testbed in Torino, in which only the data on the OneM2M platform was processed directly with some ad hoc software without any interactive dashboard. Our current results are instead obtained by the mobility tracking application running in the edge cloud.

Furthermore, [16] focused only on detecting the mobility path of single individuals in the area covered by the WiFi scanners based on a Machine Learning approach. Thus, the approach is complementary to the macroscopic view of mobility considered in this work.

## VI. Conclusions

We addressed the problem of detecting people on city streets by means of off-the-shelf WiFi scanners. In a testbed developed within the 5G EVE H2020 project, we implemented a mobility tracking application in the edge cloud by combining two VNFs that retrieve the information captured by the scanners from a OneM2M platform. One VNF is responsible to process the data, clean it and save into a local database. Another VNF is instead devoted to providing a dashboard by which users can access the data in real time through an interactive web visualization tool.

The experimental data collected span the course of 17 months and were shown to be very effective to understand the behavior and mobility of people, e.g., during work/holiday period or due to the COVID-19 restrictions, despite the coarse sampling time adopted by the scanners and the imprecise coverage information. The proposed approach is fully transparent and does not require the installation of any application in the mobile devices of the people whose presence we aim at detecting. It is also scalable and could be extended to cover whole urban areas, opening the possibility to capture macroscopic mobility trends and understand social behaviors with an unprecedented level of aggregation.

## Acknowledgments

The work has been supported by European Horizon 2020 Programme through the project 5G EVE on “European 5G validation platform for extensive trials” (grant agreement n. 815074).

## References

[1] European 5G validation platform for extensive trials. [Online]. Available: https://www.5g-eve.eu/
[2] OneM2M. [Online]. Available: http://www.onem2m.org
[3] Designed by macrovector / Freepik. [Online]. Available: http://www.freepik.com
[4] Libelium Meshlium. [Online]. Available: http://www.libelium.com/products/meshlium/
[5] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown, “A study of MAC address randomization in mobile devices and when it fails,” *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 365–383, 2017.
[6] C. Matte, “Wi-Fi tracking: Fingerprinting attacks and counter-measures,” Ph.D. dissertation, Université de Lyon, 2017.
[7] M. Nitti, F. Pinna, L. Pintor, V. Pilloni, and B. Barabino, “iABACUS: A WiFi-based automatic bus passenger counting system,” *Energies*, vol. 13, no. 6, 2020.
[8] Grafana by grafana labs. [Online]. Available: https://grafana.com/
[9] Prometheus. [Online]. Available: https://prometheus.io/
[10] Kibana and elasticsearch. [Online]. Available: https://www.elastic.co/
[11] L. Oliveira, D. Schneider, J. De Souza, and W. Shen, “Mobile device detection through WiFi probe request analysis,” *IEEE Access*, vol. 7, pp. 98 579–98 588, 2019.
[12] F. Wang, F. Zhang, C. Wu, B. Wang, and K. J. Ray Liu, “Passive people counting using commodity WiFi,” in *IEEE World Forum on Internet of Things (WF-IoT)*, June 2020.
[13] Y. Yang, J. Cao, X. Liu, and X. Liu, “Wi-Count: Passing people counting with COTS WiFi devices,” in *International Conference on Computer Communication and Networks (ICCCN)*, 2018.
[14] Yolov3. [Online]. Available: https://pjreddie.com/darknet/yolo/
[15] N. Baeta, A. Fernandes, and J. Ferreira, “Tracking users mobility at public transportation,” in *Highlights of Practical Applications of Scalable Multi-Agent Systems. The PAAMS Collection*. Cham: Springer International Publishing, 2016.
[16] 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)*. IEEE, 2020, pp. 326–330.