From Wireless Sensor Networks towards People centric sensing

Luca Filipponi
From Wireless Sensor Networks towards People centric sensing
# Contents

## Introduction

1 From WSN towards People-centric sensing  
1.1 The role of custodians  
1.2 Privacy  
1.3 Sensor availability  
1.4 Sensor context  
1.5 Mobility  
1.6 Application space

## The noise monitoring case study

2 People centric sensing for environmental monitoring  
2.1 Environmental noise pollution monitoring  
2.2 Opportunistic environmental people centric sensing  
2.3 Related work  
2.4 System overview  
2.4.1 Actors  
2.4.2 Data collection and delivery  
2.4.3 Privacy preserving data representation  
2.5 Set Covering without the Sets  
2.5.1 Greedy Algorithm for Set Cover  
2.5.2 Sketch-based Greedy Algorithm for Set Cover  
2.6 Privacy and Security  
2.6.1 Quantifying Location Privacy  
2.6.2 Enforcing security  
2.7 Experimental Analysis  
2.7.1 Algorithms  
2.7.2 Metrics  
2.7.3 Dataset
2.7.4 Experimental Results ........................................ 60
2.8 Conclusions ..................................................... 71

3 WSN for environmental monitoring .................................. 73
3.1 Data collection in WSNs ........................................... 74
3.2 System requirements ............................................. 75
  3.2.1 Hardware ................................................... 75
  3.2.2 Sampling rate .............................................. 75
  3.2.3 Data rate and latency ..................................... 76
  3.2.4 Network lifetime .......................................... 76
  3.2.5 Network topology ......................................... 76
  3.2.6 Synchronization ........................................... 77
3.3 Prototyping platform ............................................. 77
  3.3.1 Noise level measurements ................................ 78
3.4 Data collection protocols ........................................ 78
3.5 Assessment and analysis of protocols’ performances .......... 81
3.6 Conclusions ..................................................... 87

II People Centric Social Sensing ..................................... 89

4 An Opportunistic Recommendation System for Mobile resource constrained devices: Case Study .................. 91
  4.1 Mobile social networking ..................................... 93
  4.2 Related Work ................................................ 96
  4.3 Social networking over SMS messaging ..................... 96
    4.3.1 Recommending Social Relationships .................... 97
    4.3.2 Locally inferring community structure .................. 99
  4.4 Efficiently mining the social network of SMS users ....... 100
    4.4.1 Background: estimating Jaccard coefficient ........... 101
    4.4.2 Computing and maintaining contact list sketches ...... 101
    4.4.3 Exchanging sketches ................................... 103
    4.4.4 Fully decentralized recommendation of contacts ...... 103
    4.4.5 Implementation issues ................................ 103
  4.5 Experimental Analysis ....................................... 105
    4.5.1 Objectives ............................................. 106
    4.5.2 Data sets and contact graphs .......................... 106
    4.5.3 Experimental scenario for recommendations .......... 108
    4.5.4 Performance indices .................................... 109
    4.5.5 Experimental results .................................. 111
  4.6 Final considerations ........................................ 117
Abstract

Driven by the introduction of sensing capabilities into personal devices and mobile phones, many aspects originally related to wireless sensor networks, have shifted towards the involvement of users in the sensing process. People have become the focal point, and often the beneficiary of this process. Instead of sensors, placed in specific places to sample a target environment, the leveraging of human carried devices, already widespread into modern society, brings the point of sampling closer to people. Typical issues for WSNs, like energy saving and scalability lose their importance, paving the way to new application opportunities.

This thesis aims to trace the steps that have brought from wireless sensor networks to the evolution of people centric sensing, analyzing the main aspects, challenges, solutions and to put on emphasis how this new paradigm can be leveraged to perform large-scale fine-grained monitoring.

The shift towards a people centric approach is not free of challenges, among which privacy is a critical issue that can seriously undermine users participation. Dealing with human carried personal devices demands proper solutions for privacy preservation and this thesis contributes with techniques and algorithms that can provide a suitable and tunable privacy level, without affecting basic system functionalities. All these aspects are described in the first part through a case study, based on the exemplification of an application scenario, under the general context of environmental noise monitoring. The human-centric nature of this approach opens the way also to new opportunities of sensing, that enlarge their scope from physical to social. Therefore, in the second part of this thesis, we focus on the aspects related to social sensing and propose a recommendation system for mobile resource constrained devices, that opportunistically leverages SMS communication and personal informa-
tion to infer and suggest social relationships among users, still guaranteeing a proper privacy preservation. Compared with existing solutions this approach is based on a fully decentralized architecture, and proves its effectiveness using a privacy preserving representation of user’s profiles, opportunistically shared among users by means of the residual space in natural SMS messaging. This feature makes the proposed solution adoptable also by resource constrained devices. Although the techniques and algorithms described in this thesis are generally contextualized into an application scenario, they can be applied to any kind of information in typical people centric sensing systems.
Introduction

“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it”. With this sentence, Mark Weiser coined the phrase “ubiquitous computing” around 1988[126] and gave voice to new computing and communication era, that has radically transformed corporates, communities, and personal spheres. At the present time, two decades after, forms of ubiquitous information and communication networks are evident everyday’s life, in which the foreseen ratio of many computers per person has been effectively reached and continuous developments are rapidly under way to take this phenomenon an important step further, by embedding short/long range mobile transceivers into a wide array of additional gadgets and everyday items, enabling new forms of communication between people and things, and between things themselves. Under this huge framework, many technologies have been developed and concepts like RFID (Radio Frequency Identification) or WSN (Wireless Sensor Network) have been embodied into pervasive devices of everyday’s life. Focusing in particular on the latter, not more then 10 years ago, wireless sensor networks started gaining popularity and most research effort has been spent to make this technology mature to find its way out of the research world, for being employed in military applications or industries such as automotive, homeland security, medical, aerospace, home automation, remote monitoring, structural and environmental monitoring. Pushed by this innovation the original concept of computer network has been enriched with networks of small custom devices, generally tailored for specific applications and deployed, statically or not, over an area to measure physical dimensions, being part of a pervasive system able to perform tasks that were unimaginable few years before.

Parallel to this phenomenon, we have also assisted to the natural evolu-
tion of consumer electronics towards smarter devices, able to provide more and more services to their users. The introduction of sensors and transducers into these devices has definitely sealed a new definition of wireless sensor network, enlarging the original vision and opening new opportunities to leverage aspects like human mobility, personal sensing or social sensing. The use of personal devices as sensors in large scale networks, definitely represents a milestone in which users have been put in the middle of the sensing process, as well as the beneficiary of it. This new kind of approach, due to its nature, has been called “People Centric Sensing” [36]. Although traditional static wireless sensor networks still remain the most advanced and suitable technology to perform fine grained monitoring of natural or artificial phenomena, people centric sensing opened new application possibilities. Under this context, some of the main limitations of WSNs, highlighted by real world experiments, can be overtaken, but new challenges have to be faced. In particular, aspects like human mobility can be leveraged to achieve large scale monitoring but at the same time introduce potential issues, as we’ll see in chapter 1. Moreover, the introduction of personal devices, directly or indirectly shipping users’ information, into the sensing process yields to potential privacy leaks, and since the involvement of users is crucial in this new paradigm, the preservation of it and the guarantee of proper privacy levels is a must in any application.

**Problem statement**

One of the most important challenges in convincing users to adopt emerging technologies is the protection of data and privacy. Concerns over privacy and data protection are widespread and are becoming more and more crucial in this era of ubiquitous computing. Sensors and smart tags can track users’ movements, habits and ongoing preferences but at the same time concepts like data request and data consent risk becoming outdated. Invisible and constant data exchange between things and people, and between things and other things, will occur mostly unknown to the owners and originators of such data, and this issue becomes even more effective with the diffusion of new technologies and applications among masses. The fear of an Orwellian “Big Brother” raises the question about who will ultimately control the data
collected from all the information spread by people in this new concept of
network. A trivial answer is that the best owner of user’s private information
is the user itself, but this consideration leads to the conclusion that a proper
privacy preservation is demanded for people that take part in the world of
pervasive and ubiquitous computing.

To promote a more widespread adoption of the technologies underlying the
“Internet of Things”, and in particular user-centric ones, principles of informed
consent, data confidentiality and privacy must be safeguarded. Moreover,
protecting privacy must not be limited to technical solutions, but encompass
regulatory, market-based and socio-ethical considerations.

**Thesis outline and contributions**

This thesis presents the work done during the 3-year PhD program in com-
puter engineering, and mostly focuses on privacy preservation in people-centric
sensing applications. Although this is the main topic, the approach to people-
centric paradigm is subsequent to a previous research experience on static
wireless sensor networks. This experience has been instrumental to under-
stand the benefits of such technology, but at the same time to understand
how its limitations can be overtaken by leveraging human carried personal de-
vices in people centric applications. Under these premises, chapter 1 aims at
tracing the steps that have brought to PCS as a natural evolution of the orig-
inal concepts related to Wireless sensor networks. In this chapter, an analysis
of the main aspects involved in a people centric sensing scenario is presented,
together with a description of the state of the art and major research results.
The rest of this thesis is articulated into 2 parts, part I aims, through a case
study, at exploring the opportunities to leverage people carried devices for ur-
ban scale environmental monitoring, as well as static wireless sensor networks
in their typical application context. For this purpose we take as exemplifica-
tion the application scenario of environmental noise monitoring and, in chapter
2, we start from a people-centric sensing application to discuss techniques and
algorithms that can be applied to guarantee privacy preservation of personal
information. Starting from a typical architecture for people centric sensing
systems, we introduce a technique that allows the central authority to select
a subset of users whose past positions provide a good coverage of a given area of interest, without explicitly georeferencing them. To achieve this goal, we propose an efficient algorithm to solve the well known, NP-complete Set Cover problem that does not require explicit knowledge of the sets, but only uses their compact, privacy preserving representations called sketches. The algorithms we discuss are independent from data and, although we analyze them applied to georeferenced information, they can be applied to any kind of sensitive data. As an evidence of the previous background experience in the field of WSNs, and the applicability of this technology to the same application scenario, in chapter 3 we present an architecture for environmental monitoring through static sensor networks. Since requirements and constraints imposed by this technology are orthogonal to those typical of people-centric applications, in this chapter we focus on energy efficiency, which is the major aspect to deal with in the WSN context made of battery powered resource constrained devices. With the objective of evaluating the feasibility of this approach, as well as pointing out its limitations, we provide a comparison of two main MAC protocols for WSNs, based on direct measurement techniques. These two works together help to understand the main advantages and drawbacks of the respective approaches in the same application context, putting on emphasis how the involvement of human carried devices in the sensing process can solve some main issues or improve the ability for a system to scale, but with some drawbacks deriving from the people-centric nature of those devices. Part II completely focuses on People centric sensing and aims at exploring the new opportunities brought by the leverage of personal devices for sensing purposes. Social sensing is one of these and in chapter 4 we present and analyze a distributed recommendation system for resource constrained mobile devices. Since social relationships is one of the new dimensions that a PCS system can measure, in this chapter we exploit this opportunity to design a distributed system architecture, and adapt some of the techniques already introduced in previous chapter to cope with an opportunistic leverage of SMS communication. We show how it is possible to implement a privacy preserving recommendation system in a fully decentralized scenario. Furthermore we provide an experimental analysis of the techniques we use, to show their effectiveness over two major real datasets, the first deriving from a real world
experiment[49], the second extracted from Facebook’s social graph[8].

Publications

Part of this thesis has been published in the following journal articles, conference and workshop proceedings:

- Luca Filipponi, Silvia Santini, Andrea Vitaletti. Data Collection in Wireless Sensor Networks for Noise Pollution Monitoring. *In proceedings of the 4Th IEEE Intl. conference on distributed computing in sensor systems (DCOSS ’08), Santorini island.*


- Luca Becchetti, Luca Filipponi, Andrea Vitaletti. Opportunistic privacy preserving environmental noise pollution monitoring. *In Proceedings of International Workshop on Sensing for App Phones (PhoneSense), 2010*


Chapter 1

From WSN towards People-centric sensing

Driven by technological improvements of wireless communications and subsequent miniaturization of microprocessors and hardware platforms, the last decade has seen the study and development of wireless networks formed by several small resource-limited embedded devices that communicate via low-power, low-bandwidth radio with the ability to sense different kinds of environmental measures. The challenges offered by such resource poor devices and their application to real world scenarios have been in the spotlight of research and several successful field experiments showed the feasibility of these systems to provide fine grained observations of the physical world. WSNs allow Long-term unattended operation enabling measurement at spatial and temporal scales that were impractical with human observers or sparsely deployed instruments. The availability of small, battery powered, cheap devices, capable of being deployed over adverse areas and auto-organize to provide data with no infrastructural cost, has made this technology very attractive also to industries and public boards, considered that automation capability improves data quality and uniformity of measurement, while reducing data collection costs as compared with traditional human-centric methods. WSN devices are designed to operate for long periods in habitats that are inhospitable, challenging or ecologically too sensitive for human visitation. Furthermore, WSN make possible unobtrusive observation that is the key for studying natural
phenomena. While microelectronics’ evolution, following Moore’s Law, has allowed the design of tiny, low-power devices and platforms with proper capabilities, and future expectations are going to confirm the trend, this is not true for what concerns battery energy density (figure 1.1), resulting in an effective bottleneck for this technology.

Therefore, from the very beginning, research has mainly focused on the study of networking protocols that were able to cope with the typical challenges of these networks, such as energy consumption, but also with data latency and scalability. After a decade in which research in this field has been very active and a vast number of protocols (see [44, 127] for an exhaustive survey) and algorithms designed for specific applications have been presented, the application domains of WSN are continuously expanding and include wildlife and structural monitoring, target tracking, disaster management and precision agriculture [33, 61, 110, 120].

Several standards are currently either ratified or under development for
wireless sensor networks. There are several standardization bodies in the field of WSNs: the IEEE focuses on the physical and MAC layers; the Internet Engineering Task Force works on layers 3 and above. In addition to these, bodies such as the International Society of Automation provide vertical solutions, covering all protocol layer. Although a standardization process seems to be ongoing, the vast majority of products rely on non-standard, proprietary mechanisms and specifications. This is also due to the fact that these resource constrained devices, often demand some sort of “cross-layer” optimization, that yields to specifically tailored solutions, and thus, many works have been proposing vertical network protocols that cover Mean access, addressing and routing in a single stack. Despite of the original expectation, WSNs still struggle to spread outside research world, and effectively deployed industrial solutions are still rare at the moment.

On the other side, the last decade has seen the introduction of sensing capabilities into consumer electronics, and objects such as mobile-phones, originally intended to offer telephony services, have increased their scope to a wider range of possible applications. Following this trend, a new research field on People Centric Sensing has gained popularity [36], leveraging human carried devices and mobility to perform sensing tasks as large scale distributed sensor networks. Putting people and personal devices in the middle of the sensing process allows to bring sensors closer to the real beneficiary of it, and at the same time enlarges the range of sensing dimensions, from physical dimensions, typical of static WSNs, to social or personal ones. This opens the way to a wider range of possible applications “from people for people” and allows the sensing process to be integrated with human activities and information. Achieving this is not trivial, and even though the leverage of human carried devices can help in overtaking some of the characteristic limitations of static WSNs such as scalability and cost-effectiveness, it yields to some new and challenging issues to be solved specifically, related to privacy, mobility and heterogeneity. Based on these premises, the shift toward a people-centric sensing approach makes some of the typical challenges for static WSNs lose their relevance with respect to this new scenario. Starting from the devices themselves, table 1.1 depicts an overview of the main characteristics of traditional static sensor nodes together with the typical human carried personal devices.
such as a modern smartphone. This is not intended to give a comparison that could appear trivial, but it actually helps to understand how concepts like energy efficiency, multihop communication or data latency, widely used in static WSNs, now become somewhat optional, while aspects like privacy and sensor context gain importance. Resource richer human carried devices (i.e. Smartphones) as datasources at the edges of the network can be a more active part in the sampling process, performing elaboration or providing visualization other than simple sampling and data transmission that are typical in WSN motes. Dealing with smartphones rather than traditional sensor nodes shifts out the concerns about computational power, storage and battery management, if we consider the latests models that appeared on the market (e.g. IPhone 4s, HTC HD2, Samsung Galaxy S2) they are all equipped with Dual Core processors, whose performances are close to those of some years ago PCs. Battery duration is no more an issue if we think that these devices, primarily intended to serve the user with telephony and messaging, are daily recharged by owners or often in proximity of power sources. Communication can exploit multiple network interface that allow short, medium or long range communication, thus making possible to let devices directly send data to the Sink of the network, or exchange information between peers. This allows new system architectures, in which the typical WSN architecture based on many-to-one grouping behind a resource richer proxy can be revised to fulfill new scaling requirements that are still unfeasible with traditional WSNs. Regarding the sensing capabilities, the typical set of sensors include accelerometers, compass, gyroscope, camera, light sensor, but we foresee that many other will be introduced in the near future. We have also to consider that, due to standard protocols, it’s possible to extend the sensor set with additional ones, as an example the Nike+ sensor can be interfaced with Ipods and Iphones to get tracking and audio feedback when running; other projects like [6] aim at interfacing sensors like ECG or blood pressure to turn a personal device into a patient monitoring gateway. Due to the above considerations, personal devices gain the role of super-nodes (or collecting entity) of sensor networks tailored around people rather than the environment, and at the same time they act as sensor nodes in a large-scale network providing several heterogeneous services, in a distributed or centralized way.
Moving the focus from architecture to functionalities, as described in [80],
geneneral purpose people-centric sensing systems must provide the following: *application query submission, device selection, sensor sampling, and data analysis, sharing and presentation*; all of these, in turn, can be part of sequenced and dependent stages.

In the application query submission stage, a query is submitted to the system specifying at least one required sensor type (e.g., camera) and a set of conditions (i.e., sampling context) under which sampling of the required sensor should take place (e.g., location, time, physical orientation). The device selection phase aims to select the subset of nodes, among available devices, that will be tasked to meet a particular submitted application query. Due to human mobility, the main issue at this stage lies in the availability of a minimum number of sensors, with adequate characteristics, and able to accomplish the sensing task, namely defined as *sensor availability problem*.

In the sensor sampling stage, the selected device samples the sensor required by the application query if its own sampling context complies with that specified in the query. Talking about sensor context demands the definition of the role of the custodian, in fact it can be both active or passive part in the sampling process, and this can have an impact on the quality of sampled data. Custodian as active part in the sampling process (Participatory Sensing) takes place when the custodian consciously opts to meet an application request (e.g. giving inputs or simply placing sensors in a suitable way), thus producing higher quality samples. On the other side, custodian may not be involved in any of the sensing stages and thus device can be part of the sampling process in an Opportunistic way, whenever its context meets the application requirements.

In the data analysis, sharing, and presentation stage, depending on the application requirements, the sensor samples are analyzed (e.g., filtering, classification, etc.) perhaps in combination with results from other queries. The results of the analysis, and possibly the raw samples, are then returned to the querying application and may also be shared with others, depending on issues such as connectivity and privacy. Starting from these definitions, stages can be sequenced in a different order, as an example, in the system architecture presented in chapter 2, the sensor sampling is autonomous and collected data
are sent to a centralized authority, the query submission is then executed as an extemporary task, based on offline data, and thus a selection is done among collected traces to identify those which satisfy the desired parameters. On the other side, in chapter 4 we discuss an architecture in which sampling occurs opportunistically whenever a short message is received, and after analysis, results are presented whenever the results satisfy a similarity requirements (implicit query). In the following, we will discuss the key concepts and issues related to general purpose people centric sensing systems, analyzing the state of the art and the corresponding literature.

1.1 The role of custodians

The role custodians may take in a people centric sensing scenario can vary, depending on the application, between two extremes: participatory or opportunistic, as described in [80], where authors model and compare these two opposite approaches. The level of user participation has a direct impact on several other aspects such as performances of the system, privacy and data quality. Users’ participation requires people to take active part in the sensing process, incorporating them into the main stages of the sensing system, allowing to decide when and what data is shared, letting custodian to choose to what extent its data can be useful to meet application requirements. Under this approach, the custodian is conscious of its contribution to the sensing process, and which personal information are disclosed to trusted or untrusted entities, for the application purposes.

For this reason participatory systems design focus on tools and mechanisms that assist people to share, publish, search, interpret and verify information collected using personal devices, as well as social techniques to make users’ participation more attractive, especially in the bootstrap phase, when a critical mass of community appeal and engagement has to be created, as described in[46]. This approach based on users participation has been considered less challenging because most of the application support is carried on by users in a manned way, and actually much attention has been paid in the research community to applications and system based on opportunistic networking and sensing.
1.1. THE ROLE OF CUSTODIANS

<table>
<thead>
<tr>
<th></th>
<th>Sensor nodes</th>
<th>Smartphones</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>Not replaceable battery. Need for energy aware protocols. Many real deploymen</td>
<td>Often recharged battery. An excessive energy wasting can prohibit basic usage. Anyway not strict requirements.</td>
</tr>
<tr>
<td></td>
<td>ts use power cabled networks.</td>
<td></td>
</tr>
<tr>
<td>Computational</td>
<td>Usually limited</td>
<td>Rapidly increasing, dual core CPU’s in most recent models</td>
</tr>
<tr>
<td>power</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storage</td>
<td>some KBs</td>
<td>some GBs</td>
</tr>
<tr>
<td>Sensing capabilities</td>
<td>Related to physical dimensions</td>
<td>Can sense physical and social dimensions. Due to human carriage, data quality can be affected by inaccuracy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td>Single low range communication device</td>
<td>Multiple network interface, for short/medium/long range with different specs (datarate, power drain, etc)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mobility</td>
<td>Typically static</td>
<td>Follows human mobility models</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HW Heterogeneity</td>
<td>Null in the same network. Need suitable middleware/hardware for inter-networ</td>
<td>High, but easier to manage, due to official standard technologies (Wi-fi, BT, etc)</td>
</tr>
<tr>
<td></td>
<td>k communication</td>
<td></td>
</tr>
<tr>
<td>Localization</td>
<td>Need suitable protocols or hardware</td>
<td>Internal GPS based</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synchronization</td>
<td>Need suitable protocols or hardware</td>
<td>Internal GPS based</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COST</td>
<td>100 USD per node</td>
<td>300 USD per node</td>
</tr>
<tr>
<td>Commercialization</td>
<td>Research laboratories</td>
<td>Masses</td>
</tr>
</tbody>
</table>

Table 1.1: Comparison between WSN devices and modern smartphones

In an opportunistic approach, the custodian takes active part only in the deployment phase, by installing the application, setting proper parameters such as desired services and privacy level. After this initial stage, the user may not be involved in the sampling process and custodian’s device (e.g., smartphone) is utilized whenever its state (e.g., geographic location, body location,
resources) matches the requirements of the sensing task. In this scenario, the need for suitable context awareness mechanisms arise, moreover considering that, since user is not involved in the sampling activity, devices and sensors can be likely carried in a way that is not the most convenient (e.g. a pocket or purse), thus sampled data need to be correlated with their context to be analyzed (both at sampling stage or as a post processing task). The sharing of all these information opens the way to potential leak of personal sensitive information, directly or indirectly, when providing sensor or context data, thus representing one of the main issues to be faced with proper solutions. Moreover, resource management has to take into consideration that primary services like telephony and bandwidth don’t have to be affected during normal usage, since this can seriously mine the appeal of the application itself. For the opportunistic nature of the sensing task, sensors could not be always available, thus enlarging the probability to miss minimum sensor availability requirements.

While Participatory and opportunistic represent the two extreme approaches, in practice most systems place themselves in middle ground, as an example the CarTel [71] project demonstrates how both approaches can be combined together. CarTel is a mobile sensor computing system designed to collect, process, deliver, and visualize data from sensors located on mobile units such as automobiles. A CarTel node is a mobile embedded computer coupled to a set of sensors. Each node gathers and processes sensor readings locally before delivering them to a central portal, where the data is stored in a database for further analysis and visualization. In the automotive context, a variety of on-board and external sensors collect data as users drive, these sensors need user intervention to be installed and connected. For communications, CarTel nodes rely primarily on opportunistic connection to available wireless (e.g., Wi-Fi, Bluetooth), or to “data mules” such as other CarTel nodes, mobile phone flash memories, or USB keys to communicate with the portal. On the other side the Urban Sensing project at CENS-UCLA[98] tends more towards a participatory approach, in which users can optionally be active part in the system, being subscribers for data streams published by sensors. A fully participatory example is the DietSense project [103], which aims to monitor the consumption of food by the participant, requiring them to take photos of the food during meal situations to assist dietitians in monitoring patients. Manual
intervention in the review and filtering of the image data is required.

The MetroSense Project [90] demonstrates to which extent, the adoption of opportunistic approaches can help the application scaling to very large areas. Built on top of it, Bikenet [50] is an extensible mobile sensing system for cyclist experience mapping, leveraging opportunistic sensor networking principles and techniques. Bubblesensing [85] is a new sensor network abstraction in which sensing tasks are opportunistically assigned to mobile phones which deliver data a central server for retrieval by the user who initiated the task.

1.2 Privacy

Independently from the role of users, potentially dealing with personal devices leads to major issues regarding the preservation of privacy of information related to user’s data, habits, locations and, more in general, all kinds of sensitive information. We believe that concerns with data privacy are key impediments to the proliferation of such applications. Since users become part of the sensing process, but have no visibility on the whole system and actors, it is likely that people would not agree to share personal information to an untrustworthy application. A case that demonstrated sensibility of users on the disclosure of their own personal information happened in April 2011 [10], when a bug on iPhone 4 was discovered. Till that moment, iPhones had been storing and maintaining a database of Wi-Fi hotspots and cell towers to help speeding up the computation of position when requested. The official answer told that this data was sent to Apple in an anonymous and encrypted form, but the fact that many users discovered a feature they were not aware, and this feature, without explicit consensus, regarded personal information such as positioning, and possible tracking, brought some panic among Apple community. The importance of this tricky issue regarding privacy has been pointed out by Apple itself, that trying to answer users’ questions regarding this bug published “Providing mobile users with fast and accurate location information while preserving their security and privacy has raised some very complex technical issues which are hard to communicate in a soundbite. Users are confused, partly because the creators of this new technology (including Apple) have not provided enough education about these issues to date”. This situa-
tion confirmed that unrestricted dissemination of users’ sensor data results in breaches of privacy, while users want to control who may access information about themselves. Differently from the large amount of security solutions already addressed in the sensor network literature, the confidentiality of sensed data in people centric sensing goes far beyond the provision of a secure channel from the sensor node to some gateway node. Taking account of the stages that characterize the sampling process, we can simplify considering two phases: tasking and reporting.

Tasking activity, when applied to people centric sensing, can introduce some privacy leaks. As an example, let’s consider a monitoring task assigned to a sensor for a particular area, the assignment itself can help to infer the time in which that sensor has been in that place, and since sensors are related to human mobility, this can potentially allow the tracking of a person.

More in general, this issue has already been addressed by the Anonysense project[42]. Anonysense allows applications to submit sensing tasks that will be distributed across anonymous participating mobile devices, through a Task Service. It receives task descriptions from applications, performs some consistency checking (related to the carriers’ privacy requirements and the feasibility of the task), and distributes the current tasks to mobile nodes when they ask to download new tasks. Since all tasks are downloaded by nodes in a public way and then each node fulfills a subset of them, there is no clear assignment of task X to node A. Anonymous tasking is not enough to provide privacy, indeed an attacker could monitor a mobile node’s actions (accepting tasks or submitting reports), and correlate this linked history of time-location events to known carrier patterns and identify which carrier’s mobile node generated those reports. For this reason, Anonysense applies anonymous reporting through an anonymizing network to hide nodes’ location while reporting data, (e.g., by bouncing data between the anonymizing network’s nodes several times before the data goes to the database). If one organization manages all the nodes, however, a system administrator may be able to correlate routing information and infer a node’s location.

The anonysense approach requires some trusted entities in the architecture, and the problem becomes non-trivial in the absence of a shared trusted entity that can be used to sanitize the data. Moreover, since the data itself,
1.2. PRIVACY

such as GPS traces, may reveal user identity, anonymity cannot be considered the final answer to the privacy problem. In other words there is the need for techniques and architectures that guarantee privacy also in an untrusted scenario but at the same time allow to perform data elaboration, preferably not reducing data fidelity. PriSense [113] is a solution that goes in this direction, providing privacy-preserving data aggregation in people-centric urban sensing systems. PriSense relies on two components, the first component aims at additive aggregation functions. Its basic idea is for each node to slice its data into a certain number (say, n + 1) of slices before answering the query from an aggregation server. Then it randomly chooses n other nodes, called its cover nodes, to which a unique data slice is sent. Finally, each node sends to the aggregation server the sum of its own slice left and the slices received from others along with little side information, based on which the aggregation server can compute an accurate additive aggregation result. In this way, a user’s data will be disclosed only when the aggregation server and all his cover nodes collude. The second component is a non-trivial combination of the slicing and mixing technique and binary search to enable privacy-preserving Count queries and to further support a wide range of non-additive aggregation functions like Max/Min, Median, Histogram, and Percentile, all with accurate results. Starting from similar settings, in chapters 2 and 4 we propose some techniques for privacy preservation in people centric sensing, that do not require any trusted entity. The work we discuss in chapter 2 aims to support privacy in people centric sensing applications by obfuscating sensible data using minwise independent permutations, thus producing a representation of it called sketch. As we’ll see, this representation of data allows a service provider to compute relevant statistics over the data without disclosing the original content even to the service provider itself.

Since georeferenced data is one of the most sensible kind of information, many works in literature have taken this as a reference scenario and have focused on privacy in location based services[101, 135] and many Location Privacy Preserving Mechanisms (LPPM) have been proposed. As stated in [116], this research literature is not yet mature enough on the topic and, although several contributions have been proposed to quantify privacy in specific areas such as databases [48], anonymity protocols [38], anonymization networks
or RFID privacy [124], there is still the lack of a unified and generic formal framework for specifying protection mechanisms and also for evaluating location privacy. The adversary model is often not appropriately addressed and formalized, and a good model for the knowledge of the adversary and his possible inference attacks is missing. This can lead to a wrong estimation of the location privacy level of mobile users. A unified framework for location privacy has been recently proposed by [114] based on some main concepts:

- A location-privacy preserving mechanism (LPPM) acts as a noisy channel that modifies the information that is communicated from the users (as the source of information) to the adversary (as the observer/receiver),

- Users’ location privacy is maximized if the adversary cannot correctly link their location and identity over time,

- Ideally, the amount of information leakage should be minimal, while enabling users a proper use of the service. More realistically, there is a tradeoff between privacy-leakage, data fidelity and/or efficiency.

It’s important to note that many considerations related to location-privacy mechanisms can also be applied to different sensible information (e.g.: social information), and thus many techniques can be exploited also in other application contexts, as we’ll see in chapter 4. In the following we discuss a taxonomy (fig. 1.2) for the classification of LPPM that will be taken as reference in the following chapters. As a first classification regarding architectures, LPPMs can be divided into three different categories:

- **Distributed**: As described in chapter 4, they can work in a distributed way, being implemented in Mobile nodes, without the need of trusted central authority.

- **Centralized**: They rely on a trusted central entity that applies the privacy preserving mechanism before sharing data with untrusted entities. Since data transfer between mobile users and the trusted entity can be subject to various types of attack, this architecture often needs cryptography to secure this phase.
1.2. PRIVACY

- **Hybrid**: A suitable mix of the Distributed and Centralized approaches, as in chapter 2.

Another orthogonal classification can be done on the basis of the techniques used for information privacy preservation:

- **Obfuscation**: Raising information’s inaccuracy or imprecision level [47] by adding noise to it or by coarse graining them. This method can be implemented in all architectures using various algorithms. In the existing privacy preserving mechanisms, obfuscation is achieved mostly through perturbation [63, 65, 91] or generalization [17, 19, 60, 129] algorithms. The privacy preserving techniques presented in chapters 2 and 4 of this thesis mostly rely on Obfuscation through generalization.

- **Anonymization**: Altering the identity of an event in order to make it disjoint from the user that generated it and from all other events from the same user. A distinction is often drawn between “true” anonymity, where an individual is indistinguishable from all other individuals in a set, and pseudonymity, where an individual maintains a persistent identity (a pseudonym) that cannot be linked to their actual identity. Many implementations (e.g. [42]) make use of pseudonyms that can be assigned by a central entity (if trusted) or self assigned and changed in particular situations (or places called mix-zones) from which is hard (at least to a certain extent) to recover the owner of the pseudonym [22, 34, 56]. The mechanism proposed in [81] also makes use of group pseudonyms, and users exchange their group pseudonyms each other when they leave mix zones.

- **Event Hiding**: A basic method for protecting location privacy hiding information about the location of users. A subset of events is removed in the transformation process. This method is implemented mostly in distributed architectures where mobile devices refrain from transmitting information by being silent during certain time periods. Privacy sensitive users, or privacy tools (e.g., [69, 70, 74, 81]) make mainly use of this method (along with other methods). Moreover it can also be implemented in centralized architectures, indeed a service provider which
Figure 1.2: LPPM Classification Taxonomy.
follows some privacy policies, in practice, is applying this method. The mechanisms proposed in [66, 67] are examples of using event-hiding privacy tools, especially, in centralized architecture.

- *Fake event injection*: Relies on injection of fake information in the sampling process. This method can be effectively implemented in centralized architectures and mechanisms proposed in [40, 76, 78, 87] mainly employ it. Generating a trace of mixed events (real and dummy) that looks like a normal user’s trajectory enforces privacy of users with respect to eavesdropping attackers. The important drawback of this approach is the fact that an higher amount of data has to be transferred, and thus the privacy level is proportional to the communication cost. As an example, in [76] this technique is coupled with a cost reduction technique for communication.

Finally, in order to define how privacy preservation can be quantified, in literature, various metrics have been used to estimate location privacy in different scenarios. These can be mainly divided into two main categories, based on the set of the criteria that is used in each metric: *uncertainty*-based and *error*-based metrics. An Uncertainty-based metric was originally proposed by [45] and [111] for anonymous networks (known as entropy-based or information theoretic-based metrics), and by [107, 119] for database privacy (known as k-anonymity metric, where, assuming maximum uncertainty for the adversary, k is equal to the effective anonymity set size [111]). These metrics were adapted to measure location privacy in [63], and later used in many papers such as [17, 23, 59, 60, 134]. Virtually all of the various versions of uncertainty-based metrics for location-privacy, measure the adversary’s success in the presence-disclosure identification attacks. Error-based metrics are based on the adversary’s error in tracking/identifying users or disclosing their personal information: the higher is the error, the higher is the level of privacy. This category is divided into multiple subclasses:

1. Clustering-error metrics: The adversary’s goal is the clustering of the observed events into partitions. Two slightly different versions of this metric are used in [54, 65]. Note that both versions aim at measuring the success of adversary’s tracking attack.
2. Probability of error metrics: The adversary’s probability of error in finding the real identity of a user, or linking his observed events, is considered as the metric. For identification attacks, in [77] various algorithms using machine learning techniques are proposed to identify the homes of mobile users in the observed events and subsequently find their identities based on the adversary’s knowledge. Similarly, in [64] an algorithm is proposed to identify users based on their home addresses. The higher the average adversary’s probability of error is, the higher is the level of users’ location privacy in their model. In [57] the probability of error is used as the metric to evaluate users’ location privacy against tracking attacks in mix zones.

3. Distortion-based metrics: Having prior knowledge about the system, and after observing a set of events, the goal of the adversary is to reconstruct the actual trajectory of users. The distortion-based metric presented by [115] reflects how distorted the reconstructed trajectory of each user will be for the adversary. To measure the distortion, it is enough to condition the possible actual trajectories of the targets to the observed events, and compute the expected distance of the predicted location of a user with his actual location at any time instance. It is shown that this metric is superior to the other macro metrics that focus on tracking attacks, in terms of the accuracy of the metric. A set of criteria, derived from the definition of location privacy, is also proposed in [115] to compare the effectiveness of the metrics.

1.3 Sensor availability

A people centric sensing system, involving personal devices is characterized by a high degree of hardware and software heterogeneity among mobile sensing devices, this implies that there may often be a mismatch between the sensing capabilities required by application queries and the available resources on the queried sensor devices. Devices can be based on completely different hardware platform, with different computational power, storage, or communication capabilities. This last aspect is mainly mitigated by standards (e.g.: Bluetooth, Wifi, etc.) that allow interoperability as far as the standard itself is imple-
1.3. SENSOR AVAILABILITY

mented. Under these circumstances, heterogeneity regarding communication capabilities, can be described in terms of “implemented standards”. As an example, let’s consider a 3G (UMTS) and a 3.5G (HSDPA) mobile phones, they will be both able to connect to remote services, but exploiting different technologies with different available bandwidth, making the former less suitable for bandwidth demanding tasks. Another important aspect regards sensing capabilities, since devices can be equipped with different sensor sets (e.g.: accelerometers, compass, GPS, light, etc.). Due the market competition between producers, sensing capabilities of devices used to converge to the same levels in the last years, and smartphones from concurrent manufacturers, belonging to the same generation, usually are equipped with the same sensors set. Despite of this, smartphones belonging to different generations may be equipped with a set of sensors that usually has a big intersection, but it’s not equal. As an example let’s consider Iphone 3G and Iphone 3GS, even though they are produced by the same manufacturer and run the same operating system, the 3G is not equipped with compass as its successor is. Similar considerations stand for what regards software. Early ages’ mobile phones used to run very poor operating systems, providing only basic services and no expansion or reprogramability. When these phones became smarter and smarter, full featured operating system started to be put on devices, supporting different hardware platforms, in a way that reflects what happened in the personal computers’ market. The possibility for these operating system of being extended with services and applications also made possible to leverage these devices for purposes which are different from those the producer foresaw.

All the above considerations are at the origins of the so called “Sensor Availability problem” that formally refers to the difficulty in assigning application requests to mobile sensors that are suitably equipped to meet them. Due to the hardware/software heterogeneity, among all mobile sensors, it’s important to identify the subset of mobile sensors that are equipped with proper capabilities and thus are eligible for tasking under the application purposes. Once this subset has been identified, it must be assessed whether it is able to fulfill the sensing process with the desired data quality/granularity. As an example, considering a noise monitoring application in which the system requests mobile sensors to sense the sound level in a particular area, a subset of
nodes with microphone and GPS capability has to be detected, and the size of this has a direct impact on the fidelity and granularity of gathered data. The sensor availability problem has been faced under the Metrosense\cite{90} framework in which network elements are classified into architectural tiers according to available computational, communications, sensing and energy resources. Service implementations must be aware of, and take advantage of such resource asymmetry that exists between the tiers by requiring the higher capability tiers to handle tasks that consume more resources, or require a broader view of the network status. Leveraging resource asymmetry may result in sub-optimal process flow in the provision of a particular service or operation, but yields to flexibility in the sensing applications that can be supported.

1.4 Sensor context

As described in the previous section, the eligibility of a sensor for tasking mainly depends on the matching between device’s resources and requested sensing capabilities, but we have also to consider that among eligible nodes, only a subset of them can provide data with a quality that matches application requirements, depending on their context. Context awareness in people centric sensing applications is a crucial aspect to yield higher fidelity data for a particular scenario, indeed, considering that mobile sensors are human carried devices, often carried in a way that is not the most comfortable for a sensing process, the ability to recognize the context allows to infer whether a sensor can provide high data fidelity or not. As an example, both \cite{89} and \cite{75} present two approaches to monitor noise pollution involving citizens and built upon the notions of participatory sensing. They both evaluate the feasibility of using mobile phones as soundmeters in a people centric environmental monitoring process. Especially in \cite{75}, despite users’ participatory approach, the authors put on emphasis the effects of the context on data fidelity. The way in which the context has an impact on performances involves both users and technical factors (e.g. performance affected by the weather, noise interference from user movements, noise interference during user’s conversations). This impact becomes even more drastic shifting from the participatory to the opportunistic approach. From the experiences of \cite{89} and \cite{75} the need for a
suitable data filtering emerges, and thus the knowledge of the sensor context becomes crucial in order to apply suitable strategies for data collection, with the objective to satisfy the data quality requirements.

At the same time the context itself can be the object of the sensing process. As an example the CenceMe project [93] aims at exploiting physical and virtual sensors to capture user’s status in terms of his activity (e.g., sitting, walking, meeting friends), disposition (e.g., happy, sad, doing OK), habits (e.g., at the gym, coffee shop today, at work) and surroundings (e.g., noisy, bright, etc). This information is collected in order to allow injection of it into popular social networking applications such as Facebook, MySpace, and IM (Skype, Pidgin).

Context inference and recognition poses some general issues regarding acquisition, fidelity, dynamics and complexity. Works like [99] and [131] deal with these issues by proposing a collaborative learning with the objective of reducing inaccuracies in context recognition and labeling. In particular, the solution proposed by [99] takes as input data and context labels introduced by users (participatory approach) to be grouped in classes. Based on intra-class similarity, classes are then clustered into sub-classes. These sub-classes are merged based on inter class similarity and the final set of classes is defined to be taken as input by classifiers for context recognition. The effectiveness of this approach is demonstrated by means of an experimental analysis based on simulations, on the other side the authors of [131] present the results of a real case study, involving sensor-equipped taxis all over the city of Beijing. The collaborative approach is similar to the one of [99], and is organized in 2 main stages: context acquisition (local acquisition and sharing) and context management (filtering, composition and storage). Both papers demonstrate with different evaluations, how enhanced situation awareness by means of context sharing can help reducing errors in context recognition and thus improving the output of the sensing process.

1.5 Mobility

In a people centric sensing scenario, users’ aggregate mobility both enables sensing coverage of large public spaces over time and lets individuals, as sensing device custodians, collect targeted information about their daily patterns
and interactions. Envisioned solution in the field of static sensor networks are clearly untenable in terms of monetary cost, scalability or management, therefore mobility is the key factor for a system to scale in a way that allows urban-scale sensing. The sensing coverage of spaces, events, and human interactions is opportunistic in a people-centric system because the system architecture has no point of control over the human mobility patterns and actions that facilitate this coverage. Although this lack of control can translate into gaps in sensing coverage, many works ([12],[13],[11]) have focused on probabilistic coverage methods in mobile sensing, on one side providing algorithms to select the minimum set of nodes whose sampled data are sufficient to answer a query over an area of interest, on the other side analyzing lower bounds on coverage of a particular area with respect to the required data fidelity. In chapter 2 we formally define this problem and show the results of an experimental analysis aiming to evaluate the performances of proposed algorithm also in terms of achieved coverage.

Furthermore, mobility takes an important role in opportunistic networking scenarios such as [50], where mobile nodes rely on short range data transmission during occasional rendez-vous with other nodes to forward their data or, when in contact with a base station, leverage long range transmission channels (CDMA, WIFI). The unpredictability of rendez-vous makes this paradigm suitable for delay tolerant networks due to the “store and relay” strategy that has a deep impact on latency of gathered data. An outstanding work regarding the effects of mobility on networks performance has been presented by [62] further analyzed in [112] and [83]. The work presented in [62] demonstrates how mobility of node and rendezvous can be leveraged to increase the per-user throughput. This improvement can be achieved by exploiting a form of multiuser diversity via packet relaying. Starting from this theoretical aspects, [37] presents an exhaustive experimental analysis on six different traces from which emerges that the distribution of the inter-contact time seen between two devices in an opportunistic networking environment exhibits a heavy tail over a large range of value, that can be compared to a power law with a coefficient less than one. This important result is in contrast with the exponential decay assumption made implicitly by mobility models used to date in ad-hoc networking and highlights the need for new mobility models, hopefully closer
1.6 Application space

Considering People centric sensing as a natural evolution of Wireless sensor networks, deserves some additional considerations about application scenarios in which the transition toward a PCS approach can bring tangible improvements if compared to a solution based on a traditional WSN. Research on sensor networks was originally motivated by military applications, with possible scenarios ranging from large-scale acoustic surveillance systems for ocean surveillance to small networks of unattended ground sensors for ground target detection. In addition, the availability of such low-cost platforms and sensors has been exploited for many other applications, such as infrastructure security, counterterrorism, borders protection, industrial sensing, environmental and habitat monitoring, personal healthcare. During the early years of research, WSN application space has been well defined and many development have demonstrated benefits and drawbacks of this approach, nevertheless only a limited number of prototypes has effectively reached the market for being employed in real case scenarios. The main obstacles in this path lie in the fact that, due to its nature, this technology has always suffered of strict constraints imposed by the scarceness of resource that WSN devices have to face. Although electronics evolution has partially mitigated issues regarding computational power and storage, the main bottleneck still lies in battery power and short range communication. Starting from the latter, the inability of WSN devices to communicate over long distances, requires suitable multihop transportations protocols when network dimension scales over very large areas. This implies that communication cost is shared also among nodes that are neither originator nor recipient of the information, and if we think that these devices are designed to be battery powered, node lifetime can be seriously reduced since communication is the most expensive activity. Moreover, the failure of a node can potentially lead to network partitioning, thus suitable redundancy is required, raising deployment costs. All these characteristics strongly limit the ability of such networks to scale over very large areas, demanding proper countermeasures such as gateways and hierarchical architecture design.
On the other side, the ability to sense the environment has been introduced into popular consumer electronic devices, making them perfect sensors to sense the environment under a user-centric perspective. Furthermore, people-centric sensing has brought new opportunities to enlarge the application space, allowing to extend the range of sensing from physical to social dimensions. In past sections we have discussed how this transition can solve many of the above issues, while introducing new ones, here we want to focus on the application space in which these two approaches can be applied. Obviously wireless sensor networks still remain the most suitable technology for unmanned monitoring in hazardous environments, or military fields, as well as social sensing can be performed only by means of human carried devices. Nevertheless there is a slight overlap between the two application spaces represented by applications that can be developed with both approaches, that allow us to better understand benefits/drawbacks of each one. Environmental monitoring represents the application scenario in which this overlapping is more effective. As a striking example, many works regarding traditional wireless sensor networks have focused on noise pollution monitoring in urban scenarios \[108, 109\], evaluating platforms and protocols for data gathering, dealing with static networks of small, cheap resource-constrained devices. A work, following this directions is presented in chapter 3 of this thesis, and even though feasibility of this approach has been demonstrated, the inability of scaling to large-scale deployments and its cost-effectiveness remains questionable. On the other side,
new concepts related to the rise of people centric sensing have been proposed to overtake these limitations, and works like \cite{75, 89} showed how to leverage human mobility to perform environmental noise pollution monitoring in large scale urban scenarios. Taking inspiration from this new direction, in chapter 2 we exemplify a typical environmental monitoring application, and analyze the issues related to privacy in PCS, providing techniques and algorithms to perform a privacy preserving environmental monitoring in large scale scenarios.

As discussed before, the leverage of human carried devices for sensing activities has enlarged this application space to the personal and social spheres, bringing new opportunities for applications that aim at monitoring nonphysical dimensions such as social activity or personal preferences (e.g.: music, habits). For this purpose, “virtual sensors” (e.g.:\cite{93}), rather than physical ones, have been modeled as the resource from which, a PCS application can sample dimensions related to the context of the human behaviour. Starting from this premises, in chapter 4 we introduce an SMS-based lightweight recommendation system, leveraging personal mobile phones to infer social relationships among users in a community.
Part I

The noise monitoring case study
Chapter 2

People centric sensing for environmental monitoring

In this chapter we present a case study regarding environmental noise pollution monitoring as a reference scenario. In particular we take this application scenario as an exemplification to discuss and analyze the main issues related to the use of people-centric sensing paradigm for opportunistic environmental monitoring. The main objectives of this chapter are the definition of a reference architecture with a particular emphasis on privacy related issues that arise when the sensing involves personal human-carried devices sharing geolocated information. In the scenario we envision, each phone within a given area has periodically its microphone polled to get an estimation of how noisy a particular location is. Furthermore, the location of the device can be gathered via GPS if available or via Wi-Fi positioning techniques. These information are sent to a central repository or authority that exploits them to answer statistical queries regarding some geographical areas. After the definition of a typical system architecture, in the first part of this chapter, we introduce a technique that allows the central authority to select a subset of users whose past positions provide a good coverage of a given area of interest, without explicitly georeferencing them. To achieve this goal, we propose an efficient algorithm to solve the well known, NP-complete Set Cover problem that does not require explicit knowledge of the sets, but only uses their compact, privacy preserving representations called sketches. We perform a thorough experimental analysis
to evaluate the performance of the proposed technique and its sensitivity to a few key parameters using public data from real applications.

In the second part of this chapter, we will briefly show a similar solution, belonging to an earlier work, for environmental noise pollution monitoring by means of static wireless sensor networks, in order to have a clear comparison between the effectiveness of these two approaches, compared in the same application context.

2.1 Environmental noise pollution monitoring

Recent studies [52] emphasized the astonishingly severe health problems exposure to noise pollution in urban areas is responsible for, like hypertension or ischaemic heart disease. Furthermore, exposure to noise pollution can negatively affect productivity and social behavior, as well as causing mental disturbs. Considering that conservative estimations give in about 300 millions the number of citizens within the European Community that are affected by noise pollution, the Directive 2002/49/EC of the European Parliament has made the avoidance, prevention, and reduction of environmental noise a prime issue in European policy. In order to precisely assess the real dimension of the problem, the European Commission required member states to provide an accurate mapping of environmental noise exposure in urban areas like public parks, schools, hospitals, and other noise-sensitive zones. The availability of these maps will allow for a more precise analysis of the problem and for an educated design of adequate noise abatement policies. Furthermore, the creation of new noise maps at regular time intervals, will provide a wide and reliable data basis to evaluate the effectiveness of the adopted policies. While current noise maps are mostly based on sparse data and ad-hoc noise propagation models, a recent position paper by the Commission [53] has stressed that “every effort should be made to obtain accurate real data on noise sources.”

The demand for accurate data about noise exposure levels will therefore increase dramatically, as this statement makes its way into mandatory regulation.

In particular, today’s noise measurements in urban areas are mainly carried out by designated officers that collect data at a location of interest for
successive analysis and storage, using a sound level meter or similar device. This manual collection method does not scale as the demand for higher granularity of noise measurements in both time and space increases. Even if this assessment procedure is still compliant with European regulations, today’s computational models often fail to provide accurate estimations of the real noise pollution levels. Indeed, while the free propagation properties of noise generated from typical noise sources\(^1\) are well understood [25], shadowing and reflection effects hinder accurate estimation of noise levels in complex urban settings. For instance, estimated noise levels on internal buildings façades (e.g., facing a courtyard) are typically unreliable, and this inaccuracy may become critical if noise exposure data is used to drive decisions about construction planning or to elaborate local noise abatement policies.

Instead, a network of wireless sensor nodes deployed over the area of interest could collect noise pollution data over long periods of time, and autonomously report it to a central server. Moreover, since sensor nodes are typically equipped with several different sensors, they can label the collected noise data with additional information like, e.g., location values registered as the noise measurements were collected. On the other side, the introduction of sensors into consumers’ electronic such as mobile phones has increased the availability of devices that can be leveraged to perform this task, also from the point of view of the beneficiaries of it. Even though both static wireless sensor networks and human carried devices can be used to support this sensing task, these two approaches raise orthogonal issues to deal with.

In the following section we will discuss how environmental noise pollution monitoring can be addressed at urban scale leveraging opportunistic people-centric sensing.

2.2 Opportunistic environmental people centric sensing

In a recent report by IBM [72], the use of smartphones as mobile sensors is envisioned to be one of the top five innovations that will change our lives in the

\(^1\)Typical noise sources are, e.g., human activities, motor vehicles, railways, aircrafts or industrial machinery.
next years and Qualcomm Chairman and CEO Dr. Paul Jacobs predicts that by 2014, there will be over 400 million wearable wireless sensor devices on humans. In his key note at last Sensys, Alex Sandy Pentland [100] discussed how to “instrument humanity”, namely how to exploit wearable wireless sensors carried by humans to support novel and advanced services such as mapping social networks, predicting traffic patterns or classifying human behavior.

People-centric sensing goes in this direction by leveraging personal devices and human mobility to enlarge the range of action of static WSNs. Recalling from chapter 1, PCS can see an active participation of the users (participatory sensing) it can employ an opportunistic approach, in which a user configures her device to allow an application to run (subject to privacy and resource usage constraints), but may not be involved or aware of the application’s activity at any given time [80].

The challenges and constraints introduced by these approaches have been already discussed but, while these issues may fade over time or be at least partially addressed, serious privacy issues remain, since both participatory and opportunistic sensing often involve the disclosure of potentially sensitive data to the application and/or the service provider. Privacy is an important issue in the design of mobile sensing applications, as an example, referring to the environmental noise monitoring mapping scenario, in which data is autonomously collected by users, who are then expected to communicate their data to a service provider. This in turn selects the subset of users that provided data collected in the area of interest, and then computes and visualizes the map. This selection process typically allows the service provider to become aware of users’ location traces, exposing them to privacy threats. In this scenario, people are sensitive about how location data captured by the phone, needed for geotagging of noise data, is used by the system or eavesdropped by external attackers. Users need to be convinced that their privacy is safeguarded. In the following sections we present an approach to support privacy in people centric sensing applications. In particular we use sketches, namely compact and privacy preserving synopses of data, that allow a service provider to compute relevant statistics over the data without disclosing users’ sensitive information even to the service provider itself.

Roadmap. We discuss related work in Section 2.3. Section 2.4 describes the
2.3 RELATED WORK

overall scenario we envision and the main aspects of our approach. In particular, it describes the data collection and delivery process and it gives an overview of the general technique we use to produce compact and privacy preserving sketches of users’ geographical data. Section 2.5 describes an efficient algorithmic technique that allows the service provider to select a subset of data, among those received from all mobile users, that best match a statistical query over an area of interest. Section 2.6 briefly discusses the privacy robustness of our technique and outlines a general approach that can be adopted to make communication between mobile users and the service provider secure against third party attacks. Section 2.7 describes in detail an extensive experimental analysis we conducted using a large collection of GPS traces. The results obtained suggest that the approach we propose is effective and realistic. Finally, in Section 2.8 we summarize the main findings of this work.

2.3 Related work

Environmental monitoring at urban scale has been one of the main application scenarios for People centric sensing. The MetroSense project [90] is working with industry and agencies to develop new applications [50, 93], classification techniques [86], privacy approaches [42], and sensing paradigms for mobile phones [35, 85] enabling a global mobile sensor network capable of societal-scale sensing. The use of people centric sensing approach for noise monitoring has been proposed by few works. SoundSense [86] is a framework for modeling sound events captured by the microphone of mobile phones. SoundSense leverages the existing body of work on acoustic signal processing and classification and takes a systems design approach to realize one of the firsts of such systems for a mobile phone. A key design goal of SoundSense is the scalability of classification to a large population. The privacy issue is addressed in SoundSense by locally running on the phone with no server interaction. Moreover, all raw-sound traces are discarded after elaboration and cannot be recovered by malicious attackers. Similarly the NoiseTube [89] project has developed a platform for the monitoring of urban noise pollution, based on mobile phones. NoiseTube enables citizens to measure their personal exposure to noise in their everyday environment by using GPS-equipped mobile phones as noise sensors.
The geolocalized measures and user-generated meta-data can be automatically sent and shared online with the public to contribute to the collective noise mapping of cities. In this work privacy is slightly addressed by leaving users free to put their measures in the public domain, or not, so that they can be used for a scientific purpose or to build a collective noise map. The user owns his/her data. Thus for each session of measurement, or by default, he can decide to make his measures public and whether to contribute to the collective noise mapping. In this case users are exposed to potential attacks such as tracing of their position, and the responsibility to accept this potential privacy leak is left to users themselves that can give up their privacy in favour of community. The same approach has been followed by the NoiseSpy [75] project in which noise maps are built on the basis of data coming from users’ devices in a participatory way. An ongoing work in the NoiseSpy project has begun to address privacy issues providing privacy policies, which inform the user about data handling practices and serve as the basis for the user’s decision to release data. Moreover, an additional verification protocols that ensure secure data management and guarantees system integrity is under design.

More in general, out of the specific context noise monitoring, PCS systems have to be designed and implemented to protect the privacy of participants while allowing their devices to reliably contribute high-quality data to large-scale monitoring tasks. Anonysense [42] allows applications to submit sensing tasks that will be distributed across anonymous participating mobile devices, later receiving verified, yet anonymized sensor data reports back from the field. Anonysense has been designed to provide two key security properties: anonymity for the carriers and integrity for the sensed data. As far as anonymity concerns, the main goal of the authors is to avoid adversaries to de-anonymize a carrier. In our approach, also the service provider (i.e. the Report Service in [42]) is not aware of the carrier location. In other words, we could participate to a mapping service managed by Google without explicitly disclosing sensitive information to Google itself.

Privacy preservation in location based services has already been addressed by [101, 135]. In [101], accurate traffic speed maps in a small campus town are built from shared GPS data of participating vehicles, where the individual vehicles are allowed to “lie” about their actual location and speed at all
times. In our approach, data are always correct but represented in a compact and privacy preserving way (i.e. sketches). The sketching techniques we adopt in our approach allow us to guarantee a satisfactory accuracy in the reconstruction of the observed phenomenon using logarithmic space in the number of samples. Differently from [135], where data are available in clear to the intended receiver, in our work sketches allow a central authority to select relevant traces to reconstruct an accurate map, but without revealing to anybody (central authority included) relevant information on users’ positions.

2.4 System overview

In the setting we consider, our goal is to leverage users’ smartphones to sample environmental data in the surroundings of their current locations. Data collected in this way are delivered to a central authority for further processing. Using these data, the central authority can perform monitoring tasks, compute statistics about ongoing environmental phenomena in an area of interest and answer complex queries about such phenomena.

A strong issue under this approach is the preservation of user privacy, since the application has to collect and process data originating at users’ personal devices, which might at least in part be sensitive. In particular, sampled data must typically be geo-referenced to be of any utility. As a consequence, users’ movements could be easily tracked with a serious loss of privacy if these data were simply disclosed in clear. Thus, a privacy preserving representation of data is an important requirement. The above discussion implies that we have the following, general issues: i) a privacy issue, in the sense that the central authority should receive the information needed to perform its tasks, but sensitive data should be represented in a way that prevents access to sensitive information, such as geographical positions of the users; ii) a security issue, i.e., we don’t want sensitive data to be intercepted by a third party during their delivery to the central authority; iii) at the same time, data should be represented in a form that allows the central authority to manipulate the data, as to extract the information that is necessary to perform its task. We stress that this architecture, here described in the context of noise monitoring, is based on a general setting that can be applied to the vast majority of PCS
applications, as well as the techniques we present, that can be leveraged to preserve privacy in geo-located services as well as other kinds of services (see chap 4).

2.4.1 Actors

The general scenario for people centric sensing we envision involves three main actors: mobile users, central authority and system users.

- **Mobile users.** These are responsible for data collection. They participate to the service by running a monitoring application on their phones. This application can exploit on board sensors or other resources to sample data over environmental (Microphone, Accelerometers, Gyroscope, camera) or social (contacts, SMS, etc.) phenomena. These data are eventually delivered to a central authority, which is in charge for their processing.

- **Central authority.** The main goal of the central authority is to elaborate data collected by mobile users, so as to provide a monitoring service to systems users.

- **System users.** These submit queries that in general entail the computation of one or more statistics over the data collected by users in an area of interest. These can be for example the average noise or air pollution in the area of interest, an estimate of the number of people attending a social event etc.

2.4.2 Data collection and delivery

In the scenario we consider (see figure 2.1), we have $n$ users moving in an area $U$ and collecting data on one or more phenomena of interest. More precisely, at any point in time each mobile user maintains data about the set of the last $k$ positions she visited. In particular, the $\ell$-th trace generated by user $i$, denoted by $T_{i\ell}$ consists of $k$ pairs $< p_{ti}^\ell, n_{ti}^\ell >$, with $t = 1, \ldots, k$, where $p_{ti}^\ell$ and $n_{ti}^\ell$ are respectively the $t$-th user position in the trace and the corresponding sampled value of the measure of interest. Let $P_{i\ell} = \{p_{ti}^\ell\}_{t=1}^k$ and $N_{i\ell} = \{n_{ti}^\ell\}_{t=1}^k$, be respectively the set of positions (named hereafter $i$'s $\ell$-th
2.4. SYSTEM OVERVIEW

Figure 2.1: System Overview
position set) occupied by the user in her \( \ell \)-th trace and the corresponding set of samples (named the sample set hereafter).

We note that parameter \( k \) is very important for at least two reasons. First, it allows to exercise some control over the time scale of traces; in particular, smaller values of \( k \) allow result in traces that are more likely to refer to shorter time intervals, which can be useful in some applications. Even more importantly, the choice of \( k \) has an impact on performance, as discussed in detail in Section 2.7.

Also, we emphasize that \( n^i_\ell \) (and therefore \( N^i_\ell \)) is a placeholder for any set of parameters we might want and be able to measure. In particular, \( n^i_\ell \) might be a tuple of values. For example, for each position we might be interested in sampling the noise and CO\(_2\) levels, as well as record the time at which the sample was taken, so that each \( n^i_\ell \) would be a triple in this case.

### 2.4.2.1 Sample data delivery

As informally mentioned in the introduction to this section, in order to guarantee users’ privacy, \( P^i_\ell \) should not be disclosed to third parties, including the central authority, and thus it should be represented in a suitable, privacy preserving way. The sample set can be publicly available or not, depending on the application. Without loss of generality, we assume in the rest that sample sets can be publicly available. In this scenario, user \( i \) sends to the central authority the pair \((\text{Sk}(P^i_\ell), N^i_\ell)\), where \( \text{Sk}(P^i_\ell) \) is a sketch, i.e., a suitably generated compact summary of \( P^i_\ell \). We defer a thorough discussion about the way we generate sketches and about their mathematical properties to Subsection 2.4.3. For the moment, suffice it so say that the sketches we use enjoy the following general properties:

- For every \((i, \ell)\) pair, \( \text{Sk}(P^i_\ell) \) represents \( P^i_\ell \) implicitly in small space (in the order of \( 10^4 \) bytes at most) and does not allow to easily infer \( P^i_\ell \);

- Considered any area \( I \) of interest, \( \text{Sk}(P^i_\ell) \) allows the central authority to estimate the extent to which \( P^i_\ell \) covers \( I \). Note that this is achieved using only \( \text{Sk}(P^i_\ell) \), so that \( P^i_\ell \) is never explicitly disclosed to third parties.
2.4. Query answering

All users’ traces $T = T_{11}, T_{21}, \ldots, T_{i\ell}$ are made available to the central authority according to the mechanism described above. The central authority in turn collects user traces and processes them to answer queries issued by system users. Each query involves the computation of some statistical aggregate of interest over data collected by mobile users within any area $I \subseteq U$ of interest. Denote by $Q(I)$ a generic query involving area $I$. Upon reception of $Q(I)$, the central authority i) selects the minimum number of position sets $P_{i\ell}$ ensuring the maximum possible coverage of $I$ and ii) computes the statistical aggregate of interest over the data contained in the corresponding sample sets. We note that step i) above corresponds to the well-known, $NP$-hard set cover problem. The central authority uses the techniques described in Section 2.5 to perform this task. In particular, the central authority only uses the sketches $Sk(P_{i\ell})$ of users’ position sets and not the position sets themselves.

The general scenario described so far is summarized in Figure 2.1.

2.4.3 Privacy preserving data representation

As discussed before, in the application we envision, a user only sends a compact summary of her position set, from which it is hard to recover the original set. In this section we present a class of sketches [29, 30, 31] that, while compact and addressing the privacy issues mentioned above, allow the (approximate) implementation of some basic primitives on sets (such as union and intersection) that are required to implement the algorithms presented in section 2.5. In the rest of this subsection we present techniques used by mobile users’ terminals to produce compact summaries of their respective position sets.

2.4.3.1 Compact representation of sets

We only briefly outline the principles underlying the technique we propose, leaving out many theoretical aspects for the sake of brevity. The interested reader can refer to [29, 30, 31]. In the remainder of this subsection, we consider without loss of generality subsets of $[n] = \{0, \ldots, n-1\}$, for a suitable integer $n$. In our case, this means that we are regarding position sets as subset of $[n]$, where $n$ is the number of possible locations. We briefly note that standard
techniques allow us to reduce to this situation in all practical cases\(^2\).

Assume we have a family \( \mathcal{H} \) of hash functions such that: i) every \( H \in \mathcal{H} \)
produces a permutation of \([n]\); ii) if \( H \) is chosen uniformly at random from \( \mathcal{H} \)
the following holds: for every set \( S \subseteq [n] \):

\[
P[x = \arg \min(H(S))] = 1/|S|, \forall x \in S,
\]

where \( H(S) \) is the subset of \([n]\) onto which the elements of set \( S \) are hashed
and where we define \( \min(H(S)) = \min_{x \in S} H(x) \). Such a family is said \textit{minwise independent} \cite{29}. In practice, minwise independent hash functions are hard
to generate, since they require a high number of truly random bits. In this
work, we use functions \cite{28} of the form \( H(x) = ((ax + b) \mod c) \mod n \),
that excellently approximate minwise independent families. Here, \( c \) is a large
prime (e.g., the well-known Mersenne prime \( 2^{31} - 1 \)) and \( n \) is the number of
possible locations in \( U \). Finally, \( a \in \{1, \ldots, c - 1\} \) and \( b \in \{0, \ldots, c - 1\} \).

\subsection{2.4.3.2 Sketch generation}

Considered any subset \( S \) of \([n]\), we construct its sketch as follows: for \( m \)
times, we choose, independently, uniformly and at random, a hash function
from a minwise independent family. Let \( H_i(x) \) the \( i \)-th function chosen. Then
the sketch of \( S \) is \( \text{Sk}(S) = \{\min(H_1(S)), \ldots, \min(H_m(S))\} \). In our case, the
generic \( i \)-th hash function has the form described above, i.e., \( H_i(x) = ((a_i x + b_i) \mod c) \mod n \). In practice, generating such a hash function means generating
\( a_i \) and \( b_i \) uniformly at random from \( \{1, \ldots, c - 1\} \) and \( \{0, \ldots, c - 1\} \) respectively.

\subsection{2.4.3.3 Sketch properties}

The sketch generation technique we consider enjoys interesting properties that
prove extremely useful to solve the problem we consider. They are briefly
discussed below.

- **Composability with respect to set union.** Given sets \( S_1 \) and \( S_2 \),
  the sketch of \( S_1 \cup S_2 \) can be immediately obtained from \( \text{Sk}(S_1) \) and
2.4. SYSTEM OVERVIEW

Sk(S2) using the following, obvious fact: Sk(S1 ∪ S2) = \{M1, ..., Mm\}, where Mi = \min\{\min(H_i(S1)), \min(H_i(S2))\}.

- **Estimation of the Jaccard coefficient.** Another interesting property of these sketches is that they allow to easily and accurately estimate the Jaccard coefficient of two sets, a standard measure of the similarity between sets, widely used in information retrieval and a key ingredient to solve the problem we are interested in. Given two subsets S1 and S2 of [n], their Jaccard coefficient is defined as

\[
J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}
\]

It can be shown [29] that for every S1, S2 ⊆ [n], if hash function H(·) produces a permutation of [n] and is chosen uniformly at random from a minwise independent family then the following holds:

\[
P[\min(H(S_1)) = \min(H(S_2))] = J(S_1, S_2).
\]

This suggests a simple statistical estimator of the Jaccard coefficient of two sets and motivates the sketch we adopted for compact set representation. To estimate J(S1, S2), we simply consider their sketches Sk(S1) and Sk(S2) and let Cm = |\{i : \min(H_i(S1)) = \min(H_i(S2))\}|. Then, the result above and a simple high probability argument allow to show that Cm/m is an increasingly accurate estimator of J(S1, S2).

2.4.3.4 Compact representation of position sets

As already discussed in the introduction of this section, every mobile user over time submits to the central authority a sketch of her most recent trace, i.e., a sketch of the set of the last k positions she visited. The general technique adopted to produce sketches of integer subsets has been discussed in the paragraphs above. Below, we describe how this is implemented in the mobile scenario we consider.

**Hash function generation.** All mobile users will use the same set H1(·), ..., Hm(·) of minwise independent hash functions. These will be generated by the central authority and then sent to each mobile user once, i.e., the
first time she joins the application. Note that, in practice, the linear functions we use are represented in terms of a small set of parameters. For example, if we use 100 hash functions, each mobile user will need to receive 202 integer values: for the generic $i$-th hash functions, the coefficients $a_i$ and $b_i$, plus $c$ and $n$, which are the same for all hash functions. This amounts to a total of less than 1 KByte, if integers are represented using 4 bytes.

Sketch generation and update. Mobile user $i$ will generate a sketch of her $\ell$-th position set incrementally as follows: her sketch $\text{Sk}(P_{i\ell})$ is initialized $\{0, \ldots, 0\}$. Let $\{M_1, \ldots, M_m\}$ be $i$’s sketch at some point. If she moves to a new position $p$ (e.g., identified by the GPS coordinates of a new base station she connects to), then $\text{Sk}(P_{i\ell})$ is updated as follows: $M_j = \min\{M_j, H_j(p)\}, \forall j = 1, \ldots, m$. This sketch update corresponds to updating $i$’s position set as follows: $P_{i\ell} = P_{i\ell} \cup \{p\}$, thanks to the property of sketch composable with respect to set union discussed in Subsection 2.4.3.3. Note that in the above paragraph we have assumed that $p$ is an integer. In fact, positions are preprocessed, so that every GPS position is mapped onto a 32-bit integer and different coordinates are mapped onto different integers. There are several efficient ways to achieve this, the simplest of which is to regard the binary representation of a coordinate pair as an unsigned integer. In our experimental analysis, to make simulation faster, we used a dictionary to give the association between GPS coordinates and integers. Though the use of a dictionary would be unnecessary in practice, it is interesting to note that its size is in the order of $10^5$ bytes and thus perfectly compatible with the storage resources available on commercial devices.

2.5 Set Covering without the Sets

As discussed in Section 2.4, the central authority serves queries submitted by system users. The generic query $Q(I)$ requires the central authority to compute some statistic over the data sampled in an area $I$ of interest. To accomplish this task, the central authority uses the sampled data received from the mobile users. We remind the reader that these are in the form $(\text{Sk}(P_{i\ell}), N_{i\ell})$, where $\text{Sk}(P_{i\ell})$ is a sketch of the $\ell$-th position set of the $i$-th user and $N_{i\ell}$ is the set of values sampled in each of the positions of $P_{i\ell}$. So,
the main problem is finding a collection of positions sets (i.e., a collection of $P_{i\ell}$’s) that together cover area $I$. For the sake of efficiency, this collection should have the smallest possible cardinality. Once such a collection has been found, the statistic can be estimated by considering, for every position set, the corresponding sampled data $N_{i\ell}$. Two issues arise here: i) finding a minimum cardinality collection of position sets that together cover $I$ is an instance of the $NP$-hard set cover problem; ii) the central authority does not know the $P_{i\ell}$’s explicitly, but only their sketches $\text{Sk}(P_{i\ell})$. We address these issues in the following two subsections.

Before doing this, we remind the reader that in the classical minimum set cover problem [125] we are given a set $I$, taken from a universe $U$, and a collection $T = T_1, T_2, ..., T_n$ of subsets of $U$. The pair $(U, T)$ is called a set system. The goal is to compute a sub-collection $T' \subseteq T$ which covers $I$ with minimum cost, namely using the smallest number of sets in $T$. In our setting, each $T_\ell$ is a position set.

### 2.5.1 Greedy Algorithm for Set Cover

The best known algorithm for the minimum set cover problem is the greedy algorithm summarized in Figure 2.2 [125]. Without loss of generality, in the pseudo-code the set $I$ to cover is assumed to coincide with the universe $U$. If this is not the case, we simply replace each set $T_\ell$ of the set system with $T_\ell \cap I$. The algorithm is very simple: it maintains a collection $C$ of sets that will belong to the cover. Iteratively, in each step it selects the set in $T - C$ that covers the largest number of still uncovered elements of $U$ (lines 3 and 7) and adds it to the collection (line 6).

Since our purpose is to give a sketch-based version of the greedy algorithm above and since it seems hard to compute the sketch of the difference of two sets from their respective sketches, we slightly modify Algorithm Standard-Greedy, by replacing $|T \cap (U - C)|$ with $|(T \cup C) \cap U|$ in steps 3, 4 and 7. Maximizing the former quantity is equivalent to maximizing the latter, as proved in the following:

3In order to make the pseudo-code more readable, we slightly abuse notation, since we regard $C$ as both a set of sets (the set cover) in lines 1 and 9 and as the union of the sets that form the cover in lines 3, 4, 6 and 7. Analogous considerations hold for Algorithm 2.3.
Algorithm Standard-Greedy

Require: set system \((T, U)\)

1. \(C = \emptyset\) (\(C\) contains identifiers of sets in set cover)
2. \(\hat{T} = T\)
3. \(\hat{T} = \arg\max_{T \in \hat{T}} |T \cap (U - C)|\)
4. while \(|T \cap (U - C)| > 0\) do
5. \(\hat{T} = \hat{T} - \{\hat{T}\}\)
6. \(C = C \cup \{T\}\)
7. \(\hat{T} = \arg\max_{T \in \hat{T}} |T \cap (U - C)|\)
8. end while
9. return \(C\)

Figure 2.2: Greedy Algorithm for Set Cover.

**Fact 1.** In Algorithm 2.2, maximizing \(|T \cap (U - C)|\) is equivalent to maximizing \(|(T \cup C) \cap U|\).

*Proof.* Consider the generic iteration of Algorithm 2.2 and assume the partial cover is the same for both versions of the algorithm at the beginning of the iteration. This is clearly the case during the first iteration, if both versions of the algorithm break ties the same way. For every set \(T \in \hat{T}\) we have:

\[
(T \cup C) \cap U = (T \cap U) \cup C = T \cap (U - C) \cup (T \cap C) \cup C = (T \cap (U - C)) \cup C
\]

where the first equality follows since \(C \subseteq U\), while the third follows since \(T \cap C \subseteq C\). Since \(C\) is fixed (it is the partial set cover computed at the end of the previous iteration), maximizing \(|(T \cup C) \cap U|\) is equivalent to maximizing \(|(T \cap (U - C))|\). \(\square\)

Another important fact is the following:

**Fact 2.** Maximizing \(|(T \cup C) \cap U|\) is equivalent to maximizing the Jaccard coefficient between \(T \cup C\) and \(U\).

The proof of this fact is obvious and follows since \(T, C \subseteq U\).
2.5.2 Sketch-based Greedy Algorithm for Set Cover

We next describe PP-Greedy, a sketch-based version of Algorithm Standard-Greedy. The main difference is that in PP-Greedy, set operations have been replaced by simple operations on the corresponding sketches.

In particular, in lines 5 and 11 of figure 2.3, following Fact 2, we choose the set \( T \), such that the (estimated) Jaccard coefficient between \( T \cup C \) and \( U \) is maximized, since this set also maximizes \(|(T \cup C) \cap U|\). Recalling the discussion in Subsection 2.4.3.3 this is, up to approximations, the set \( T \) such that \( \text{Sk}(T \cup C) \) and \( \text{Sk}(U) \) agree on the largest possible number of components. In the pseudo-code of Figure 2.3, \( Eq(U, C \cup T) = \sum_{i=1}^{m} X_i \), where \( X_i = 1 \) if the \( i \)-th components of \( \text{Sk}(C \cup T) \) and \( \text{Sk}(U) \) have the same value 0 otherwise.

Algorithm PP-Greedy

Require: Sketch \( \text{Sk}(T_i) \), for \( i = 1, \ldots, |T| \), \( \text{Sk}(U) \)
1: \( E = 0 \)
2: \( C = \emptyset \) (\( C \) contains identifiers of sets in set cover)
3: \( \text{Sk}(C) = \{\infty\}_{i=1,\ldots,m} \)
4: \( \hat{T} = T \)
5: \( \hat{T} = \arg\max_{T \in \hat{T}} Eq(U, C \cup T) \)
6: \( \hat{E} = Eq(U, C \cup \hat{T}) \)
7: while \( \hat{E} > E \) do
8: \( E = \hat{E} \)
9: \( \hat{T} = \hat{T} - \{\hat{T}\} \)
10: \( C = C \cup \hat{T} \)
11: \( \hat{T} = \arg\max_{T \in \hat{T}} Eq(U, C \cup T) \)
12: \( \hat{E} = Eq(U, C \cup \hat{T}) \)
13: end while
14: return \( C \)

Figure 2.3: Privacy Preserving Greedy Algorithm for Set Cover.

2.6 Privacy and Security

The next two subsections discuss the privacy and security aspects of the approach we propose. In particular, Subsection 2.6.2 describes and discusses a simple protocol to enforce security in the framework we consider in a cryptographic sense. Though this latter aspect is not the focus of our work, we think
it might be of interest and it also demonstrates the flexibility of the approach we propose and its compatibility with state-of-art cryptographic techniques.

2.6.1 Quantifying Location Privacy

Since Mobile users disclose their locations to possibly untrusted entities, or may unwillingly expose private information to malicious eavesdropping entities over the wireless channel, it’s necessary to define an attacker model to quantify the privacy achieved by the proposed approach. We take [116] as a reference framework for privacy quantification of our approach, framing it into the family of Location Privacy Preserving Methods (LPPM) based on obfuscation by means of precision lowering. In order to evaluate our approach as an LPPM, we must model the adversary against whom the protection is placed. In our reference scenario we consider two kind of adversaries: an external attacker eavesdropping the obfuscated traces sent from the user to the central authority, and the central authority itself; recall that a unique characteristic of our system is that the central authority should be able to compute relevant statistics on data without knowing the users’ positions.

2.6.1.1 Types of attack

Since the most powerful attacker is clearly the central authority (it knows the hash functions and all the system parameters), in the following we consider the case in which an untrusted central authority performs an attack aimed at recovering users’ traces from their sketches. We next discuss the amount of location information an attacker can recover from a sketch. We assume in the remainder of this paragraph that the attacker knows the hash functions used to produce sketches of the position sets and the sketch $\text{Sk}(P_i \ell)$ and size of a position set $P_i \ell$ she intends to recover as accurately as possible. The basic operation an attacker can perform is the following: given a position $p$ and $\text{Sk}(P_i \ell)$, check whether or not $p \in P_i \ell$. To this purpose, the attacker computes $H_j(p)$, for $j = 1, \ldots, m$. Setting in the remainder of this paragraph $S = P_i \ell$ and $|S| = s$ for the sake of brevity, the following cases can occur (see Subsection 2.4.3):
i. $H_j(p) < \min(H_j(S))$ for at least a $j$. In this case, the attacker can conclude that $p \notin S$;

ii. $H_j(p) = \min(H_j(S))$ for at least a $j$. In this case, the attacker can conclude that $p \in S$;

iii. $H_j(p) > \min(H_j(S))$ for all $j$’s. In this case, $p$ is a false positive (i.e., $p$ does not belong to $S$ but it behaves as if it did, since $H_j(p) > \min(H_j(S))$ for all $j$’s) with some probability that depends on $s$ and $m$, as we see further.

Note the important fact that from case ii above, the adversary can claim membership in $S$ for at most $m$ positions in $U$. For all other positions, it can either conclude that they don’t belong to $S$ (case i)) or nothing (case iii)).

In the remainder, we make the worst case assumption that resources are not an issue for the attacker and it can perform the above described basic operation for every potential candidate position in the universe $U$. We discuss below how to make the probability that case iii occurs large enough that, even under a brute force attack using the approach described above, a large fraction of positions in $U - S$ are false positives. As a result, if $|S|$ is sufficiently larger than $m$, the attacker may be able to recover at most $m$ of the user’s locations with certainty, but a large fraction of all possible positions will be potential candidate members of $S$, thus making the data recovered by the adversary extremely noisy and of little use.

To this end, we next study the probability that, given a position $p \notin S$, $H_j(p) < \min(H_j(S))$ for at least a $j$. To this purpose, we assume that the hash functions are minwise independent, though this in practice is not the case. The paper [28] discusses how the functions we use closely approximate the ideal minwise behaviour. From the definition of a minwise independent family (see Subsection 2.4.3) we have:

$$P[H_j(p) < \min(H_j(S))] = \frac{1}{s + 1},$$

where the probability is taken with respect to the initial, random choice of the $H_j(\cdot)$’s and where we remind the reader that we are assuming $p \notin S$. Next, for every $p \notin S$, we define the binary variable $X_p = 1$ if $p$ is a false positive,
$X_p = 0$ otherwise. We also define $X = \sum_{p \notin S} X_p$. Notice that:

$$P[X_p = 1] = P \left[ \bigcap_{j \in [m]} H_j(p) > \min(H_j(S)) \right] = \left(1 - \frac{1}{s+1}\right)^m,$$

where the first equality follows from the definition of case iii above. As a consequence, we can conclude:

$$E[X] = \sum_{p \notin S} P[X_p = 1] = |U - S| \left(1 - \frac{1}{s+1}\right)^m,$$

where we remind the reader that $U$ denotes the universe of all possible positions. Now, for a fixed constant $0 < \delta < 1$, note that we have $E[X] \geq \delta |U - S|$, whenever

$$\left(1 - \frac{1}{s+1}\right)^m \geq \delta.$$

This happens as soon as

$$s \geq \frac{2m}{\ln \delta} - 1.$$

This result follows from simple calculus after observing that:

$$\left(1 - \frac{1}{s+1}\right)^m \geq e^{-\frac{2m}{s+1}},$$

and imposing that the right-hand side be at least $\delta$.\footnote{The inequality above follows since $e^{-2x} \leq 1 - x \leq e^{-x}$ for $0 \leq x \leq \frac{1}{2}$. We choose $x = 1/(s+1) \leq \frac{1}{2}$ in our case.} This result tells us that, as soon as the size of position sets is large enough (a constant number of times the number of hash functions used), the expected number of false positives can be extremely high.

For example, if we set $\delta = 1/e$, we obtain for $s$ the condition $s \geq 2m - 1$. Under this condition, the expected number of false positives is at least $\frac{1}{e} |U - S|$. Considering that $|U| >> |S|$ in practice, it follows that a large fraction of all possible positions are false positives. Assuming that the attacker is able to recover the at most $m$ positions of $S$ that achieve the minima of the hash
functions used to generate $\text{Sk}(S)$, recovering the missing $s - m$ ones may be a non trivial task, given that in expectation, at least $\frac{1}{e}|U - S|$ positions will be perfectly equivalent candidates. It is clear that knowing a subset of the positions in a trace may help prune many unlikely positions based on geographical proximity to known positions, but this strategy becomes less and less effective as the size of the position set grows and in any case, it is not possible to distinguish false positives from positions that actually belong to $S$.

2.6.2 Enforcing security

In previous sections we have described a privacy preserving, sketch-based representation of position sets. As discussed in the previous Subsection 2.6.1, this technique prevents an external attacker or the central authority itself from recovering significant portions of the original set. Still, we have seen that an attacker (or the central authority) could still recover a portion of a user’s position set using essentially a brute force approach, at least under the assumption that the attacker knows all hash functions used to produce positions set sketches (this assumption holds for the central authority but it might be unrealistic for an external attacker). Furthermore, as discussed in section 2.4, we outlined how another general issue regards security, namely the risk for sensitive data to be intercepted by a third party during their delivery to/from the central authority (for example, the parameters of the hash functions to be used). Sketches offer some degree of privacy preservation but still some minor privacy leaks in this technique can be exploited to grasp some information about the set. For this reason, we discuss here the main aspects of an additional protocol that can be transparently applied as a black box to the sketch technique, to enforce privacy and ensure data security. Let us consider two sets $A$ and $B$, each holding a vector of length $m$. For the sake of simplicity, let’s denote each party with the name of the set she owns ($A$ and $B$). In our application to the computation of the Jaccard coefficient, the vectors will be the sketches of the respective position sets. Assume that $A$ and $B$ wish to compute the number of positions $i$ for which $A[i] = B[i]$ without revealing any additional information on the vectors. Coming back to the application context, we have two parties $A$ (Mobile User) and $B$ (Central authority), each with a private set of positions. Namely, $A$ holds a user’s position set $P_{\ell \ell}$, while
$B$ holds an area $I$ of interest. $A$ and $B$ wish to compute the Jaccard coefficient $J(P_i, I)$ (the key step of the $\text{PP-Greedy}$ algorithm in Figure 2.3) in order to perform the set cover. We will describe a protocol that uses an additively homomorphic encryption scheme $(E; D; K)$ like Paillier cryptosystem (see [97] for further information).

**Homomorphic encryption scheme.** Let $(E; D; K)$ be a homomorphic encryption scheme and assume that the message space for a public key $pk$ returned by the key generator algorithm $K$ on input security parameter $m$ is $\mathbb{Z}_p$ for some integer $p$ of length $m$. The following additive homomorphic properties hold:

1. the product of two ciphertexts is a ciphertext for the sum of the plaintexts; that is, for all messages $a; b \in \mathbb{Z}_p$ and public keys $pk$, we have $D(E(pk, a) \cdot E(pk, b), sk) = a + b$;
2. raising a ciphertext for message $a$ to power $r$ gives a ciphertext for $r \cdot a$; that is, for all $r \in \mathbb{Z}_p$ we have that $D(E(pk, a)^r, sk) = r \cdot a$.

**The protocol.** The protocol can be described as follows:

1. A picks a pair of public and secret key $(pk, sk)$ for encryption scheme $(E, D, K)$ by running the key generator algorithm $K$ on input $1^m$; for $i \in [n]$, $A$ computes encryption $a_i = E(pk, A[i])$ of $A[i]$; $A$ sends $pk$ and $(a_i)_{i \in [n]}$ to $B$;
2. for $i \in [n]$, $B$ computes encryption $b_i = E(pk, -B[i])$ of $-B[i]$, picks random $r_i \in \mathbb{Z}_p$ and sets $c_i = (a_i + b_i)^{r_i}$. Notice that by the homomorphic properties of $(E, D, K)$, $c_i$ is a ciphertext for $r_i \cdot (A[i] - B[i])$. Therefore if $A[i] = B[i]$, then $c_i$ is an encryption of 0; otherwise $c_i$ is an encryption of a random element of $\mathbb{Z}_p$. $B$ randomly permutes the $c_i$’s and sends them to $A$.
3. A decrypts the $m$ ciphertexts received from $A$, counts the number $s$ of ciphertexts that are an encryption of 0 and sends $s$ to $B$.

**Properties of the protocol.** We make the following simple observations:

- **Correctness.** The value $s$ computed by the protocol is the number of indices $i$ for which $A[i] = B[i]$, with probability exponentially close to 1.
- **Privacy of
the input. Each of $A$ and $B$ gets no information on the other party’s vector, besides what can be obtained from the output of the protocol. For $A$, this can be easily seen by exhibiting a probabilistic polynomial-time simulator $S$ that, for all vectors $A$ and $B$, on input vector $A$ and the number $s$ of positions in which $A$ and $B$ coincide (but not vector $B$) outputs $A$’s view of the protocol. Similarly, we can construct a simulator for $B$.

Obviously, the Jaccard coefficient can be computed by applying the above protocol to the characteristic vector of the two sets. The protocol will then run in time linear in the size of the underlying universe set. A much more efficient protocol is instead obtained by running the above protocol with each party holding as an input the sketch of PositionSets computed using the same sequence of random (or min-wise independent) permutations. Depending on security requirements this protocol can be used as a black box, since it doesn’t have impact on the performances of the sketching techniques we discussed.

2.7 Experimental Analysis

In this section, we present and discuss the results of extensive experimental analysis aimed at assessing the performance of our approach, in particular the performance of the $PP$-Greedy algorithm and how it compares to the Standard-Greedy algorithm, whose behaviour it tries to imitate without explicitly knowing the position sets of the user, but only their respective sketches. We first investigate the effect of increasing the number of hash functions used to calculate sketches. We then evaluate the performance of the algorithms when varying the sizes of the position sets (i.e. $P_{d_i}$) and of the area of interest $I$. This section is organized as follows: Subsection 2.7.1 briefly recalls the behaviour of the two algorithms we consider. Subsection 2.7.2 describes the metrics we define for the comparison of the algorithms, Subsection 2.7.3 describes the dataset we used for our experiments, and finally Subsection 2.7.4 analyses the performance of the algorithms.

Notation. For the sake of brevity, in the rest of this section we denote by $Gr$ and $PP$–$Gr$ respectively the Standard-Greedy and the PP-Greedy algorithm. Assume the overall number of position sets is $s$. For ease of notation, we assume an arbitrary order of the position sets (for example, the order in which
their sketches were received by the central authority) and we denote them by
\( P_1, \ldots, P_s \). We also let \( P = \{ P_1, \ldots, P_s \} \).

### 2.7.1 Algorithms

The Greedy algorithm \( Gr \) receives in input the area of interest \( I \) and the
position sets of the users. Note that every user may provide a different num-
ber of position sets. \( Gr \) provides as output a set system \( P_{Gr} \subseteq P \) that
approximates the minimum cardinality set cover of \( I \). The PP-Greedy al-
gorithm \( PP-Gr \), instead of the set \( P \), receives in input the set of sketches
\( Sk_P = (Sk(P_1), \ldots, Sk(P_s)) \), and it also provides as output a set \( P_{PP-Gr} \subseteq Sk_P \)
that approximates the minimum cardinality set cover of \( I \).

Note the following: i) in both cases the output of the algorithm is simply
a collection of identifiers of the position sets that provide a cover for \( I \) and
not the sets themselves. This information will allow the identification of the
traces to obtain in order to compute the statistic of interest over \( I \) and of the
mobile users that possess the data. The difference is that, while \( Gr \) needs the
position sets to perform this task, \( PP-Gr \) only uses their sketches; ii) \( Gr \)
provides the best coverage possible of area \( I \), though its output might not be
a cover of minimum cardinality, since \( Gr \) is an approximation algorithm for
an \( NP \)-hard problem. The only reason why the coverage of \( Gr \) may not be
100% is that some positions in \( I \) might not be covered by any position set, as
indeed is the case with the dataset we consider.

**Remark.** Note that we only compare our approach with the standard greedy
heuristic. We believe this is not a limitation, since the standard greedy heuris-
tic provides the best possible approximation of the optimal solution that can be
achieved by a polynomial time algorithm. Improving over the greedy heuristic
entails considering approaches (such as Linear Integer Programming) that can
have exponential computational times and are in our opinion not very realistic
in the scenario we consider, in which the size of the input can be very large.

### 2.7.2 Metrics

We evaluate the performance of \( Gr \) and \( PP-Gr \) with respect to three metrics.

- The *cardinality* of output i.e., the number of position sets used to cover
2.7. EXPERIMENTAL ANALYSIS

the area of interest,

- The coverage of the output, intended as the fraction of positions in the area of interest that are covered by the output.\(^5\)

- The error, defined as the fraction of positions in the output which are not in \(I\).

As an example consider the following sets \(I = 1, 3\), \(P_{Gr} = 1, 2, 3, 4\). In this case the cardinality is 2, the coverage is 100% and the error is 50%.

2.7.3 Dataset

![Google Earth's representation of GeoLife traces](image)

Figure 2.4: Google Earth’s representation of GeoLife traces

The input for our experiments has been obtained from a real public dataset published by the Microsoft GeoLife project [132, 133]. The dataset contains GPS trajectories collected by 165 users in a period of over two years (from April 2007 to August 2009). To the best of our knowledge, this is the largest publicly available dataset that summarizes a broad range of users’ outdoor

\(^5\)We calculate the coverage of \(P_{PP-Gr}\) considering the set of corresponding positions.
movements, including routine movements such as driving to work or back home, but also entertainment and sports activities, such as shopping, sightseeing, dining, hiking, and cycling. Therefore, the dataset can be used in many research fields, such as mobility pattern mining, user activity recognition, location-based social networks, and location recommendation. This dataset contains approximately 22 millions of position records, mostly concentrated in the area of Beijing (China), as results from figure 2.4.

2.7.4 Experimental Results

2.7.4.1 Role of Hash function number on the PP-Greedy performance

Recall that, as observed at the end of Subsection 2.4.3.3, it is easy to prove that the fraction of positions on which the sketches of two sets agree is an increasingly (wrt the number of hash functions) accurate estimator of the Jaccard coefficient. In this section we evaluate to which extent the accuracy of the estimator increases with the number of hash functions. To this purpose, we consider a reference area of interest $I$ to be covered and we compare the results of both algorithms on multiple runs and with an increasing number of hash functions used by $PP-Gr$. The results are averaged over 10 runs of both algorithms, for each of the reported number of hash functions, keeping the same $I$ for all experiments. Since the coverage achieved by the $Gr$ algorithm is the highest possible, this algorithm is used as a baseline to evaluate the coverage of $PP$-Greedy. Figure 2.5 describes the coverage achieved by both algorithms with a number of hash functions used by $PP-Gr$ that goes from 100 to 3500.

As expected, the performance of $PP$-Greedy in coverage increases with the number of hash functions used. In particular the $PP$-Greedy line in figure 2.5 shows an asymptotic behaviour approximating the Standard-Greedy one. The plot can be divided into 2 parts: when the number of hash functions $m < 1000$, the coverage significantly increases from a very low coverage with 100 Hash ($\sim 0.23$ compared to the 0.38 of the Greedy), to a value of 0.34 with 1000 Hash functions. In the second part, the increase of coverage is smaller, with less than 0.1 every 500 additional Hash functions.

On the other side, figures 2.7 and 2.6 respectively show the values of error and
2.7. EXPERIMENTAL ANALYSIS

Figure 2.5: Coverage achieved by both algorithms with varying number of hash functions for PP-Greedy

Figure 2.6: Cardinality achieved by both algorithms with varying number of hash functions for PP-Greedy
Figure 2.7: Error achieved by both algorithms with varying number of hash functions for PP-Greedy

cardinality for the same experiments. While cardinality follows an asymptotic behaviour as well as coverage, the error tends to remain high irrespectively of the number hash functions.

This can be explained as follows: when the number of hash functions is small, as observed before, the resolution of sketches and the ability to estimate similarity between sets is small, which results in a higher error rate. As the number of hash function increases, the accuracy in set similarity estimation improves and this brings to an increase in coverage and cardinality. However, improving accuracy in similarity estimation implies that a number of sets with a marginal overlap with the area of interest are included in the final solution, thus increasing the error. Furthermore, we note that the number of hash functions is an upper bound on the cardinality of the solution. This is trivially due to the fact that in each iteration, if the PP-Greedy algorithm adds a new position set $P_d$ to the partial solution, than it has to be the case that it strictly increases the number of entries of the partial solution’s sketch that agree with the corresponding entries of $\text{Sk}(I)$ (see the description of algorithm PP-Greedy in Figure 2.3 and Subsection 2.5.2). Of course, this can happen at
2.7. EXPERIMENTAL ANALYSIS

The results described above suggest that the larger the number of hash functions we use, the better the performance of PP-Greedy. On the other hand, we also point out that a number $m$ of hash functions generates a sketch consisting of $m$ integers, for an overall size of $4m$ bytes. Recalling that the application scenario we refer to can involve mobile devices that communicate opportunistically, available bandwidth has to be considered a scarce resource. Under these circumstances, larger sketches can provide solutions of better quality, but they can be in contrast with these network constraints.

Observing the figures, beyond 1000 hash functions, the marginal increase in performance is less significant. For this reason, in the following experiments we consider 1000 hash functions as a good trade-off between the size of the sketches, which is in this case limited to 4KB, and the quality of the approximation of the Standard-Greedy algorithm given by PP-greedy.

2.7.4.2 Role of the $P_i$’s and $I$ on PP − Greedy performance

As described at the end of the previous section, we consider the PP-Greedy algorithm using 1000 hash functions and we analyze its performance as the size of the area of interest $I$ and of the position sets $P_i$ vary. The objective of this analysis is to investigate the relations that exist between these two

Figure 2.8: Different areas $I$, with varying size, used for the experiments

most $m$ times.
variables in order to maximize the performance of PP-Greedy.

The reference area, namely the world $U$ considered for our experiments is the city of Beijing. This area has been split into smaller areas $I$ by dividing at each step the reference area by two as shown in figure 2.8. The resulting areas used to evaluate the impact of the size of $I$ on the performance of the algorithms are summarized in the following:

- 1 Area with size $\sim 71000$ discrete GPS locations ($I \equiv U$),
- 4 areas $I$ of size $\sim 17900$ discrete GPS locations,
- 16 areas $I$ of size $\sim 4480$ discrete GPS locations,
- 64 areas $I$ of size $\sim 1180$ discrete GPS locations,
- 256 areas $I$ of size $\sim 300$ discrete GPS locations,
- 1024 areas $I$ of size $\sim 90$ discrete GPS locations.

On the other side we set the size of the position sets used to perform the Set-cover of $I$, namely the size of the $P_i$’s, to the following values: 10, 50, 100, 150, 200, 500, 1000, 1500, 2000, 2500, 3000, 3500 samples.

**Remark.** As noted at the beginning of this subsection, we used 1000 hash functions in the experiments that we present below. On the other hand, we considered positions set sizes above but also well below this value. It is clear that using sketches larger than the size of the position sets they represent has little sense in terms of efficiency. The only reason why we also considered sizes less than 1000 for position sets was to analyze the interplay between this parameter and the size of $I$, as summarized for example in Figure 2.9.

The system has to be designed considering a complex tradeoff among accuracy, efficiency and privacy: while increasing the number of hash functions used clearly improves accuracy, the ratio between the size of position sets and this parameter plays a crucial role. The larger this ratio, the more efficient the system of course, since we are representing positions sets using smaller and smaller numbers of bits. Furthermore, a higher ratio between these two
Figure 2.9: Coverage achieved by PP-Greedy algorithm, as a function of the average size of $P_i$ and $I$

parameters improves privacy, as discussed earlier. On the other hand, increasing the size of positions sets reduces the granularity of the users’ traces (for a given area $I$ of interest, larger position sets are more likely to include a larger number of irrelevant positions) and thus negatively affects accuracy.

The experimental results discussed in the following, are obtained averaging the results of 10 runs for each possible combination of $I$’s and $P_i$’s sizes.

**Coverage.** Figure 2.9 shows the values of coverage obtained in the experiments as a function of the size of the $P_i$’s for each area possible size of the area of interest $I$. As already pointed out, the coverage of $Gr$ is the highest possible and it does not depend on these parameters. Its value is 0.38. On the contrary, the value achieved by $PP - Gr$, significantly changes as the sizes of the $P_i$’s and/or $I$ change.

Worst performances are obtained when the sizes of $I$ and the $P_i$’s differ significantly, due to the fact that the resulting Jaccard coefficient is fairly low. Indeed, if $|P_i| \ll |I|$, then $J(P_i, U) = \frac{|I \cap P_i|}{|I \cup P_i|} \approx \frac{P_i}{I} \ll 1$. This implies that the
CHAPTER 2. PEOPLE CENTRIC SENSING FOR ENVIRONMENTAL MONITORING

Figure 2.10: Greedy Cardinality

Figure 2.11: PP-Greedy Cardinality
2.7. EXPERIMENTAL ANALYSIS

Figure 2.12: Greedy Error

Figure 2.13: PP-Greedy error
number of minima that are common to both the sketches $\text{Sk}(P_i)$ and $\text{Sk}(I)$ is low, and thus the number of sets not included in the solution is high, resulting in a low value of coverage. On the contrary, when $|P_i| \simeq |I|$, we have $J(P_i, I) = \Theta(1)$ and consequently more sets will be included in the final solution. This is also confirmed in Figure 2.9, in which position sets of smaller size are more suitable to cover small areas of interest and vice versa. Apparently, the best performance are obtained when $|I| = |U| = 71000$ and $|P_i| = 3500$. However, this is the limit case when the area $I$ of interest is the whole world $U$ we consider, namely the whole city of Beijing in the experiments. In this case, the error is zero, because all the elements in any $P_i$ are necessarily part of the world we are considering. For this reason, we do not consider this particular case in the discussion that follows. The best coverage is obtained at the boundaries of the graph in Figure 2.9, when $|P_i| = 10$ and $|I| = 300$ or when $|I| = 17920$ and $|P_i| = 3500$. In those regions, the best performer is about 50% better than the worst one. On the contrary, the region in the middle of the graph, for sizes of $|P_i|$ between 100 and 200, is characterized by differences between best and worst performers of only 15%, but it is also the region where the coverage is lower and is about half the best coverage.

The performance of the system in terms of coverage is optimized when is possible to tune the sizes of the $P_i$’s according to the size of $I$ or vice versa. However, the size of the $P_i$’s has an impact on the security (see section 2.6.1) and on the amount of required resources (i.e., memory and bandwidth) of the mobile devices. Since the size of a sketch is independent of the size of $P_i$ from which the sketch has been generated, it turns out that smaller $P_i$’s generate a greater number of sketches that have to be stored and sent either establishing a new connection or opportunistically exploiting available ones. As an example, consider a mobile device collecting samples every minute, if the size of the $P_i$’s is 10, this device will send 6 sketches every hour, while if the limit is 60, it will send only a sketch every hour.

**Cardinality and Error.** We now focus on the other two metrics to evaluate the performance of our system, namely cardinality and error. In this case, also the performance of the Standard-Greedy algorithm depends on the size of the $P_i$’s and $I$. Figures 2.10, 2.12, 2.11 and 2.13 depict the cardinality and error values for both $Gr$ and $PP - Gr$ algorithms. These four plots show
2.7. EXPERIMENTAL ANALYSIS

a common behaviour for both algorithms. As expected, the bigger the size of the $P_i$'s, the lower the number of sets selected by the algorithm, namely the cardinality. Furthermore, areas of interest $I$ of bigger size are always characterized by solutions of higher cardinality (number of position sets in the set system).

Figure 2.14 shows two lines obtained as the average of the values shown in Figures 2.10 and 2.11. In this figure, the trend of the two algorithms is confirmed to be similar, but it is now evident that $PP - Gr$ always selects a lower number of sets in its solution.

Similar considerations apply to the error metric, as shown in Figures 2.12 and 2.13. The larger the size of the $P_i$'s, the larger the error for both algorithms. Indeed, position sets of big size and only marginally overlapping with the area of interest, marginally contribute to improve coverage, but they cause substantial error increases, because most of their positions are outside the area of interest.

Thus, it follows that $P_i$'s of small size allow to perform the set cover with a finer granularity and thus decrease error; on the other hand, it is worth to analyze how an excessive small granularity impacts on the error for the PP-Greedy algorithm.

Analogously to what we have done for the cardinality, Figure 2.15 depicts the average error. The error of $PP - Gr$ is constantly lower than the error of the Standard-Greedy algorithm, except for the first point of the plot. The reason for this discrepancy, has to be found in the intrinsic limit of the sketch-based approach to estimate similarity between small sets (i.e. position sets) and sets of much larger size (the area of interest). In fact, when the size of the $P_i$'s is 10 (i.e. the smallest size of the position sets), $PP - Gr$ has a higher average error. Comparing Figures 2.12 and 2.13, it appears that the higher average error in this case is essentially due to input instances in which $|I|$ is relatively small and comparable to 10, so that $PP - Gr$ has a non-negligible probability of detecting position sets that have small, but non empty intersection with $I$.

From the above experiments we have learnt some important lessons about the performance of the PP-Greedy algorithm. First of all, as a general consideration, the performance strongly depends on the number of hash functions used:
Figure 2.14: Average cardinality for all the I areas considered.

Figure 2.15: Average error for all the I areas considered.
2.8 Conclusions

This work presented and analyzed a novel approach to support privacy in people-centric sensing applications, based on the use of compact, privacy-preserving synopses of user traces. In our system, the selection process is based on sketches of the location traces rather than the location traces themselves. Based on this representation, an efficient algorithm to perform query matching over the set of collected data was discussed. This algorithm solves the well-known, NP-complete Set Cover problem without requiring explicit knowledge of the sets, but only using their compact, privacy-preserving sketches. Furthermore, the amount of information mobile devices need to send to the service provider to allow the selection process is modest, in the order of $10^4$ bytes for maximum accuracy in realistic cases, as our experimental analysis suggests.

The goodness of this approach will further be examined in chapter 4 where...
it will be adapted to preserve privacy in an SMS-Based recommendation system, in a context characterized by different requirements and constraints, as a confirmation that the proposed technique can be exploited in several application domains such as the analysis of social patterns, as well as environmental monitoring. As an example, in the scenario presented in chapter 4, user profiles are described by feature vectors and the goal is to leverage the proposed techniques to estimate similarities between profiles without explicitly using the profiles themselves, with the objective to give friendship recommendations in a distributed way.
Chapter 3

WSN for environmental monitoring

Previous section has described how, with people-centric sensing, it becomes possible to sense and collect information at the human level. Thus, the environment is sensed exactly where its semantics are important: at the human, rather than on a wall like traditional WSNs do. Moreover involving users in the sensing process can allow a system to scale to some extent that is unimaginable with traditional sensor networks. As a drawback, some major issues like sensor context and availability can lower the granularity and data fidelity in the sampling process, while potential privacy leaks can seriously undermine the diffusion among users community. In this section we present an earlier work than the one presented in section 2.2, that aims at evaluating the use of static sensor networks for fine-grained environmental noise monitoring. This work helps to have a clear comparison between the effectiveness of these two approaches, compared in the same application context. In particular, in the following we analyze how the typical requirements can be fulfilled by resource poorer devices and select a suitable hardware/software solution in the WSN field, then in section 3.5 we perform an experimental evaluation of two MAC protocols, representative of two main classes: slotted and CSMA-based. We evaluate these protocols under the aspects related to power consumption since energy-shortage represents the main burden when dealing with static sensor nodes, due to the fact that battery replacement is often unfeasible. This
analysis, as in few other examples, provides direct power consumption measurements, and discusses results in relation to the reference scenario. The objective is to evaluate the advantages and drawbacks in using static wireless sensor networks for urban-scale environmental monitoring, and show how some of the main limitations can be overtaken with a people centric sensing approach.

\section*{3.1 Data collection in WSN}

Driven by a huge plethora of potential application scenarios, research on wireless sensor networks preponderantly evolved in the last decade, and several successful field experiments showed the feasibility of these systems to provide fine-grained observations of the physical world \cite{33, 61, 110, 120}. In the context of pollution monitoring, wireless sensor networks represent an instrument that can provide data at a scale and accuracy that was unimaginable, though desirable, just a few years ago. Furthermore, the increasing attention and concerns of the public opinion and policy makers towards environmental issues is pushing the large scale collection of accurate pollution data, whose analysis can foster a deeper understanding of pollution phenomena and support the design of local and global action plans.

In this section, we focus on the high potential wireless sensor networks have to facilitate and improve the monitoring of noise pollution, which is currently considered “one of the main environmental problems in Europe” \cite{52}. The peculiar characteristics of noise propagation and the concrete measurements requirements, apparently make wireless sensor networks a perfect tool to be used in this context.

To understand the feasibility of wireless sensor networks to be operated as noise monitoring systems, we first distilled the main requirements the application poses on system design and that our prototype must eventually comply with. As a further step towards the realization of a noise monitoring network, we experimented with different hardware platforms and evaluated their performances. To enable the thereby selected wireless noise sensors to reliably deliver the captured data to a central sink, we then implemented and analyzed different data collection protocols and evaluated their performances in
our specific application scenario. In particular, we considered the CTP (Collection Tree Protocol) and the DMAC protocols [55, 84], and measured their performances both in terms of energy consumption and latency, to eventually select the most suitable for a noise pollution monitoring network.

3.2 System requirements

Given these premises on the problem of measuring environmental noise, we now would like to go into further details and analyze the specific requirements a wireless sensor network must comply with to be used for noise pollution monitoring. While some of them are shared also with other technologies, requirements regarding energy and topology are technology-dependent and deserve a specific analysis in this application scenario.

3.2.1 Hardware

Average loudness levels over different periods of time (hours or weeks) are typically used as noise pollution indicators [53, 109], whose computation requires measuring acoustic pressure levels using an adequate microphone and signal conditioning circuitry. The high sampling rate required to properly capture acoustic pressure (> 30kHz) appears prohibitive for resource poor sensor nodes. However, commercially available platforms are able to support the required sampling rate, as long as scheduling with radio communication is properly managed. Nevertheless, to overcome this problem the most suitable solution consists in delegating sampling and signal processing to dedicated hardware and let the actual sensor node only deal with communication and possibly optimization of data collection through adequate aggregation mechanisms. In this way, nodes can directly sample noise level values, which requires sensibly lower sampling rates.

3.2.2 Sampling rate

Currently, punctual measurements of noise indicators over short periods of time are actually collected on the field, and are then extrapolated and used to feed computational models. Since this coarse sampling granularity is the major reason for inaccurate estimations of noise pollution levels, the European Community explicitly recommends to increase measurement granularity in
both time and space whenever possible. For the purpose of noise mapping, the spacing between assessing points is required to be about 3 meters, which represents the required spatial sampling rate (i.e., the physical spacing between sensor nodes). Specific types of noise sources like some industrial activities or construction sites typically produce high noise peaks that could be lost if too coarse averaging takes place. To enable computation of noise indicators as well as detection of noise spikes, which at most extends only over few seconds, a high enough temporal sampling rate must be adopted. Any coarser average of noise levels can then be easily reconstructed from this fine-grained data collection.

3.2.3 Data rate and latency

Collection of noise data does not require sensor readings to be immediately reported to the sink. Therefore, latency in packet delivery is well tolerated and won’t be a critical parameter for optimizing network performances. Still, all collected samples must be reported to the sink, therefore the data rate may depend on the actual sampling rate. In our current prototype we have a constant data rate (1 packet per second), even when the sampling rate increases, since readings can be aggregated in a single packet.

3.2.4 Network lifetime

Since noise indicators’ values may fluctuate strongly depending on the time of the day and on the day of the week (working day or weekend), measurements must cover this variability by extending at least over few weeks. If variability over different seasons is of interest, the network installation may be required to stand for several months or to be re-deployed several times in different seasons.

3.2.5 Network topology

For the purpose of noise monitoring, the assessment points used to measure noise levels shouldn’t change during data collection. This implies that nodes location won’t change over time, and therefore that the physical topology of the network won’t vary during network operation. Nevertheless, due to the spatial distribution of the nodes, reporting data to a central sink may require multi-hop communication, resulting in typical convergecast traffic pattern.
3.3 PROTOTYPING PLATFORM

3.2.6 Synchronization

Noise readings taken at different locations from different nodes must be ordered over a global timescale for proper processing and visualization. Since the specific network topology and data collection protocol may introduce a variable and unbounded latency on data delivery, an adequate synchronization mechanism must be adopted to allow for a correct time ordering of the readings that reach the sink.

3.3 Prototyping platform

To select an adequate platform to perform noise level measurements, we tested the feasibility of two different hardware solutions. We considered the Telosb/TmoteSky platform equipped with either the SBT80 multi-modality sensor board available from EasySen\(^1\) or with a custom-made noise level meter. The custom-made noise level meter directly outputs noise level readings expressed in dB and provides a good data accuracy (total error less than 3dB), thereby complying with our requirements. As described in [109], the solution using the TmoteSky equipped with the EasySen sensor board was rapidly abandoned due high sampling rates required to properly capture the acoustic signal that represent a significant computational overhead for this resource poor sensor platform. Therefore, we selected the customized hardware as the best solution for setting up a first prototype, since the computational intensive operations can be delegated to the dedicated hardware, while the node itself can save computational resources for other tasks like communication and data processing. As an example, in the following we report some exemplary measurement results obtained using our custom made noise level meter.

<table>
<thead>
<tr>
<th>range</th>
<th>urban area</th>
<th>construction site</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fixed</td>
<td>adaptive</td>
</tr>
<tr>
<td>below 40 dB</td>
<td>5.90</td>
<td>24.90</td>
</tr>
<tr>
<td>40 dB to 50 dB</td>
<td>72.30</td>
<td>43.22</td>
</tr>
<tr>
<td>50 dB to 60 dB</td>
<td>17.90</td>
<td>14.78</td>
</tr>
<tr>
<td>60 dB to 70 dB</td>
<td>2.90</td>
<td>11.08</td>
</tr>
<tr>
<td>70 dB to 80 dB</td>
<td>0.05</td>
<td>2.66</td>
</tr>
<tr>
<td>above 80 dB</td>
<td>0.03</td>
<td>3.30</td>
</tr>
</tbody>
</table>

\(^1\)www.easysen.com
3.3.1 Noise level measurements

To test the reliability of our noise-sensing node, we performed measurements in two different scenarios: an urban area close to a traffic road and a construction site. Table 3.1 reports cumulative statistics from a representative measurement session. We collected noise level readings using both a fixed sampling rate of one reading per second and an adaptive sampling rate in which the sampling interval is halved (with 1s and 100ms as upper and lower bounds) whenever sampled noise levels exceed a fixed threshold. We introduced this adaptive sampling strategy after observing that data collected at the construction site with a fixed sampling interval seemed to depict a less noisy environment than in the urban area, contradicting our expectations. For instance, table 3.1 shows that the number of samples below 40dB is considerably bigger at the construction site than at the urban site (39.90% vs 5.90%). The reason for this counterintuitive behavior lies in the typical spiky nature of noise generated by “industrial” sources like a construction site. For instance, the use of a hammer produces short and periodic spikes, thereby increasing the actual average noise level. However, if the sampling interval is too coarse to catch these spikes their contribution won’t enter in the global average. To cope with this problem, we introduced the adaptive sampling interval, which appears to better capture the actual noise levels at the construction site. Indeed, table 3.1 shows that measurements taken at the construction site with the adaptive sampling rate clearly lie in the upper part of the noise scale (32.52% of values lie between 60 and 70dB and 22.23% are above 80db). Regarding the urban area, the analysis of table 3.1 shows a quite regular behavior of noise levels, with most of the samples (72.30%) lying between 40 and 50 dB (typical values for a quite residential area), with the influence of the nearby traffic road represented by only few (several seconds long) spikes in correspondence with vehicles passing-by.

3.4 Data collection protocols

In a WSN for environmental noise monitoring sensor nodes periodically report collected data to a central sink, which acts as a gateway between the network and a data acquisition system. In this typical convergecast scenario, common
to many environmental monitoring applications [88, 110], particular care must be taken in the choice of an appropriate data collection protocol that properly trades-off performances (i.e., network throughput and latency) with the overall energy consumption.

The design of energy-efficient data access control received considerable attention within the wireless sensor networks research community and an excellent overview on this topic is provided in [44]. Periodic environmental monitoring (PEM) represents one of the most popular application scenarios for WSNs [32], [24], [88]. In PEM deployments, sensor nodes are scattered over a large area and periodically collect and report sensor data to a central sink. Examples of measured quantity include temperature, electric conductivity, or moisture levels. As for most application scenarios for WSNs, the main requirement regards the lifetime of the network, that is hopefully designed to last several years because replacement of, or intervention on, sensor nodes is unfeasible, thus nodes must rely on their own energy supply, which often consists of common AA batteries. It is important to note that although WSNs are envisioned to encompass thousands or millions of nodes [16], [128], actual PEM deployments are build upon networks of tens or hundred, since scaling up to thousand of nodes deployed over a urban-scale scenario would be unfeasible in terms of manageability and cost. Despite of the huge plethora of works regarding energy-aware protocols for WSN, a commonly agreed taxonomy sees the classification into three main categories: slotted, random-access, hybrid.

In this work, with the objective of evaluating the feasibility of using WSNs for environmental noise pollution monitoring, we selected for comparison two of the most common protocols specifically designed for convergecast scenarios, the CTP (Collection Tree Protocol) [55] and the DMAC [84] protocol, the first belonging to random-access, the second to slotted protocols category. Since no implementation of the DMAC protocol was available at the time this work has been done, we implemented it on TinyOS 2.x [3]. To better support our experimental evaluation, provided in section 3.5, we now briefly describe the two collection protocols and their main features.

CTP [55] is a collection protocol included in the standard release of the TinyOS operating system, built upon BoX-MAC. BoX-MAC-1 [94] is a random-access protocol that is part of the standard low-power MAC of TinyOS. It is
based on the concept of Duty-cycle, thus each node running BoX-MAC-1 wakes up periodically from the sleep mode and checks for channel activity. This mechanism, called Low Power Listening (LPL), is a power saving technique that allows to move the major costs of radio communication from receivers to transmitters by avoiding idle listening, which is known to be the main source of energy wasting. A node implementing LPL awakes periodically to sample the radio channel for incoming transmissions, which begin with a preamble long at least as much as the receivers sleeping interval, followed by the actual payload. The length of the preamble ensures the receivers’ ability to catch the actual data. As we will show in the following section, the LPL mechanism allows for high energy savings, but as a counterpart significantly increases the latency in packet delivery. On top of BoX-MAC-1, CTP uses beacon messages for building and maintaining a routing tree, and data messages to report application data to the sink. The standard implementation of CTP consists of three main logical software components: the Link Estimator, the Routing Engine and the Forwarding Engine. The Routing Engine takes care of sending and receiving beacons as well as creating and updating the routing table. The Forwarding Engine forwards data frames. Each transmitted data frame is acknowledged at the link layer, enhancing the reliability of the protocol. Furthermore, the FE implements a duplicate-detection mechanism and it has the ability to detect and repair routing loops. The Link Estimator mainly responsible for determining the inbound and outbound quality of a communication link. In our experiments we measured CTP power consumption and latency both enabling and inhibiting the LPL option. In the following, we will refer to this two options as $CTP_{LPL}$ and $CTP_{NoLPL}$, respectively.

As detailed in section 3.2, the environmental noise monitoring application scenario requires the network to adequately timestamp the collected data, so that a global ordering can be reconstructed at the sink. To this scope, we added basic synchronization functionalities to CTP, by implementing, at the application level, the well known TPSN (Timing-sync Protocol for Sensor Networks) algorithm [58].

The DMAC [84] data collection protocol for convergecast communication achieves bounded latency while contextually guaranteeing low power consumption. It is based on a time frame in which slots are assigned to group of nodes
belonging to the same level in the hierarchy. Nodes implementing DMAC can wait for incoming data, transmit packets or sleep, depending on the level they belong to in the routing tree (in which the sink occupies layer 0 and the leaves layer $n$). Along the multihop path from the leaves to the sink, nodes wake up following a simple scheduling: during period $i$ ($i \in [0, n-1]$), nodes at layer $n-i$ transmit their packets, nodes $n-i-1$ are ready for reception, and all the remaining nodes keep sleeping to save energy. While transmitting, nodes at a specific level $i$ use a contention based mechanism to access the channel, therefore making DMAC a kind of hybrid protocol, which uses a TDMA (Time Division Multiple Access) global scheduling but has contention-based medium access control during transmission phases. To wake up during the foreseen time slots sensor nodes need some form of synchronization, which we provided by implementing the TPSN algorithm. The characteristics of the DMAC protocol allow to easily bound the latency as a function of both the length of the periods $i$ and the number of layers $n$. On the contrary, in CTP the latency is an unbounded function of the uncoordinated schedule of the sleeping intervals.

### 3.5 Assessment and analysis of protocols’ performances

To analyze the energy consumption of the two data collection protocols under consideration, we connected a Tmote sensor node to the Rhode & Schwarz dual-channel analyzer/power supply NGMO2 [104], an instrument that can accurately measure the current drain associated with all the states of the nodes (i.e., idle listening, transmission, reception, and sleep) in each phase of the communication protocol. We then measured the node current drain during transmission and reception, for both the CTP and DMAC protocols, using the same experimental set-up exploited in [123]. This simple setting, a single-hop network made of two nodes, allowed us to gain a clear understanding of all the major energetic aspects involved in nodes communications and to determine upper and lower bounds on nodes energy consumption. A more realistic experimental setting, considering multi-hop topologies and the effect of collisions, will be considered in further investigations. We would like to point out at this point that measurements of power consumption in wireless sensor
networks typically rely on indirect measurement methods, such as counting the number of transmitted packets or CPU duty cycles, which provide limited accuracy as pointed out in [79]. Our work aligns with few other examples (e.g., [68, 79, 123]) in providing direct power consumption measurements.

Figure 3.1(a) shows the behavior of a node running the \( CTP_{\text{NoLPL}} \) protocol, characterized by the transmissions spikes, occurring every second, whose width is proportional to the length of the transmitted packet. While in idle listening, the node drains about 19mA, which is clearly the major energy waste cause, considering also that the total average current drain is about 19.5 mA. The periodic current drain raising of 1mA, producing the square wave in the plot, originates from periodic activities of the CPU. Enabling the LPL option allows to dramatically reduce the power consumption of the CTP protocol, as shown in figure 3.1(b). The current drain while the radio is in sleep mode is again limited to few periodic CPU activities. Every 250ms (the value we set for nodes sleep period), the node wakes up the radio and samples the channel, as foreseen by the LPL mechanism. If it overhears a transmission on the medium, it keeps the radio active until reception is successfully completed, otherwise switches-off the radio immediately. The spikes in figure 3.1(b) are associated to the sampling activity, while one complete reception cycle is clearly visible in the first segment of the plot. The length of the transmission phase is considerably longer than the one observed for \( CTP_{\text{NoLPL}} \), since LPL moves the cost of communication from the receiver to the transmitter. Nevertheless, under the same traffic load of one packet per second, the total average current drain of the \( CTP_{\text{LPL}} \) protocol is only 5mA, which represents an energy saving of 75% with respect to the 19.5 mA spent on average by the \( CTP_{\text{NoLPL}} \). The \( CTP_{\text{LPL}} \) protocol can achieve even lower average current drains using longer sleep intervals that, however, will also increase packet latency. For both the \( CTP_{\text{NoLPL}} \) and \( CTP_{\text{LPL}} \), the measured current drain values represents lower bounds on the protocol’s energy consumption, since in the considered experimental setting no collisions and packet forwarding occur.

For the DMAC protocol, the lower bound on energy consumption simply corresponds to the energy required to transmit (receive) a single packet per period. Determining the upper bound is less trivial and requires some additional considerations. Nodes running the DMAC protocol can, at each time
3.5. ASSESSMENT AND ANALYSIS OF PROTOCOLS’ PERFORMANCES

(a) CTP’s energy drain with no LPL.

(b) CTP’s energy drain with LPL (sleep interval 250ms).

(c) DMAC’s energy drain during transmit slot.

(d) DMAC’s energy drain during receive slot.

Figure 3.1: Energy drain
instants, be in sleep, receive or transmit state. The power consumption in sleep state is negligible, so we excerpt it from the current analysis. To provide an upper bound on DMAC’s power consumption, we then forced nodes in transmission state to transmit for the whole duration of a period, and consequently nodes in receive state to remain active also for the entire length of a period. Figure 3.1(c) shows the current drain of the node in transmission state, in which the average current consumption is about 20.5mA. The average energy consumption of the node during reception is about 21.5mA, as shown in figure 3.1(d). This evidence aligns with the figures reported in the CC2420 radio chipset datasheet [39]. An estimate of the average energy consumption, would require to take into consideration the time spent in sleep state. In particular, if we consider a 10-layers network, with communication periods of 100ms, the DMAC’s time-weighted average energy consumption is approximatively the same as for $CTP_{LPL}$. In general, reducing the number of layers makes DMAC’s average energy energy consumption increase and (and latency decrease), while the opposite is true if the number of layers increases.

After evaluating performances in terms energy consumption, we also analyzed the behavior of the CTP and DMAC protocols in terms of latency, the results are depicted in figure 3.2. We determined lower bounds on latency considering a simple chain topology with 3 hops and a packet size of 30
3.5. ASSESSMENT AND ANALYSIS OF PROTOCOLS’ PERFORMANCES

bytes. For the $CTP_{NoLPL}$ protocol latency grows linearly with the number of hops, and measures about 10ms per hop, that is the average processing time for the forwarding of data in a multihop fashion with no channel contention. Changing topology by introducing more nodes can only lead to a performance degradation with an increase of data latency (and energy wasting).

Enabling the LPL option makes latency increase significantly, since nodes in transmission must wait receiving nodes to wake up. Latency values for the $CTP_{LPL}$ protocol are therefore bigger than for the $CTP_{NoLPL}$, and grow linearly with the amplitude of the LPL sampling interval. The lower is the duty cycle, the higher energy saving, but the higher is the latency. In this particular example, with a sampling interval of 250ms the angle between latency plots of $CTP_{NoLPL}$ and $CTP_{LPL}$ is slightly less than 45 degrees. As it’s possible to see, for the specific experimental setting, the DMAC protocol shows approximately the same behavior of $CTP_{LPL}$.

On the basis of these results it’s possible to state that there is the need for a trade-off between energy consumption and data latency. Even though latency is not a key factor in this application scenario since data are gathered for statistical purposes and, thus, an higher latency yielding to a better energy saving is sustainable, it puts a spotlight over the issue of scalability of WSNs in more general scenarios. Wireless sensor networks can provide fine grained data, but on the basis of the previous experiences, and those described in literature, scaling towards networks of thousands of nodes, deployed over an urban area seems still unfeasible due to the performances degradation of most of the state-of-the-art multihop protocols, but also to the effective manageability of it. A network of static battery powered sensors is expensive to be installed, and is expected to properly work for years. On the other side, due to low range communication capability, that demands proper multihop data forwarding, the cost of data shipping is shared among nodes that belong to the multihop path, thus potentially reducing the lifetime of entire network. The multihop data forwarding itself, that was a must in the early ages of these resource poor devices, now can result in a bottleneck since it potentially exposes data flow to single points of failure, potentially leading to a network partitioning. These aspects make WSN technology suitable in small-medium scale applications but the cost-effectiveness of it in urban-scale scenarios is
still questionable.
3.6 Conclusions

In the first part of this chapter we have described a people centric sensing application for environmental noise pollution monitoring. In this context we have given particular focus to privacy preservation that is a general issue for all applications involving users’ personal information. Based on the centralized reference architecture, we have proposed an algorithm that enables the central authority to answer queries by means of users’ data, without compromising their privacy. In the second part of the chapter, based on early experiences on static sensor networks, we have discussed an application of them in the same scenario of noise pollution monitoring. Moving the point of sensing from people to infrastructures, this work has faced different issues, and while the concept of privacy preservation has lost its meaning, energy management and network performances have gained importance. In this chapter we have experienced that Wireless sensor networks can provide a cheap and flexible infrastructure to support the collection of fine-grained noise pollution data, which is essential for the preparation of noise maps and for the validation of noise pollution models. The main drawback of this approach is that urban-scale monitoring becomes unfeasible due to many aspects like performances, energy, manageability and cost. The natural evolution of these premises, in a urban scale scenario, goes towards a people-centric sensing approach, in which the user is the focal point of the sensing process, and often the beneficiary of it. The leveraging of personal devices and their mobility allows large-scale monitoring, partially relieving the burden of management from service providers because users, as owners of their personal devices, indirectly act as caretakers of their sensors. Moreover, the use of already widespread devices as sensors, drastically reduces deployment costs, while the appeal of an application gains importance to convince users to be part of the process. In this context privacy is a serious issue that can seriously undermine users participation and the adoption of this approach. The techniques we have proposed can preserve privacy in the application scenario we considered, but, as we will show in the following chapter, can be adapted to support privacy also in other people centric sensing context like social sensing. On the other side, in people centric sensing for environmental monitoring, due to human unpredictable mobility, sensor availability in a desired time/space might be inappropriate for the required service level
of a sampling task, or available sensors could be in a situation that doesn’t allow the required data quality in the sampling activity. The entirety of these considerations suggests that, in this particular application scenario both PCS and WSN approaches seem to complete each other, allowing one to overtake the limitations of the other. Furthermore, as discussed at the end of chapter 1, the application spaces of these two approaches go far beyond this particular intersection, confirming WSN as the most suitable technology to perform sensing in hazardous scenarios (e.g. volcano monitoring, military fields, etc.) or, more in general, in non-people-centric contexts. On the other side, due to its nature, PCS in the only approach able to perform social sensing and all strictly human related tasks.
Part II

People Centric Social Sensing
Chapter 4

An Opportunistic Recommendation System for Mobile resource constrained devices: Case Study

In this second part of the thesis, we would like to explore the new opportunities introduced by the leveraging of human carried devices into a people-centric sensing process. Due to their nature, these devices allow the sensing over dimensions that go beyond purely physical ones (sound, position), and enlarge the capability of sensing of social parameters such as relationships or interactions. Under these premises, modern smartphones are the most suitable device to perform this task since they represent some sort of appendix to the custodian itself. We can classify the sensing capabilities of a smartphone, dividing its sensors into two categories: physical and virtual. While physical sensors (e.g.: microphone, camera, GPS, etc.) produce samples based on electronic transducers, a virtual sensor can be considered a source of samples derived from non-physical information such as social relationship, social activity or, trivially, the list of contacts/friends of a given person.

The leveraging of personal devices as sensors has enlarged the spectrum of sensing dimensions to some extent that was unimaginable with traditional sensor networks, as a consequence aspects related to user contexts or social
activity have become somehow “measurable”.

The interactions and social relations among users have been studied by many generations of social psychologists, however, it is still hard for social scientists to capture fine-grained data about phenomena of this kind and to find the right means to facilitate interaction. Social sensing by means of human carried devices has gained the spotlight in the social networking research field, and works like [105] demonstrate it. As an example, both Facebook and Twitter integrate the possibility of automatically geo-locating users and posts, which is somehow a form of integration between pervasive sensors (the human carried device) and social ones (the posts themselves). The CenceMe project [93] is an example of how it’s possible to leverage pervasive sensors to inject sensing presence into popular social networking applications allowing for new levels of “connection” and implicit communication between friends in social networks. Smartphones represent an ideal pervasive sensing platform to conduct social interaction studies and to close the loop by providing feedback to the users; an example of the application of it to office environments can be found in [102]. While mobile phones offer a fantastic platform for harvesting long term and fine grained data, they also pose challenges, and once again, privacy is the main issue to deal with. In this chapter we start from these premises to propose a fully decentralized approach for recommending new contacts in the social network of mobile phone users. With respect to existing solutions, our approach is characterized by some distinguishing features. In particular, the application we propose does not assume any centralized coordination: it transparently collects and processes user information that is accessible in any mobile phone, such as the log of calls, the list of contacts or the inbox/outbox of short messages and exchanges it with other users. This information is used to recommend new friendships to other users. Furthermore, the information needed to perform recommendation is collected and exchanged between users in a privacy preserving way. Finally, information necessary to implement the application is exchanged transparently and opportunistically, by using the residual space in standard short messages occasionally exchanged between users.
4.1 Mobile social networking

Mobile social networking is an emerging trend. eMarketer forecasts [1] that mobile social networking will grow from 82 million users in 2007 to over 800 million worldwide by 2012. In most mobile communities, mobile users can create their own profiles, make friends, create and participate in chat rooms, hold private conversations, share photos and videos. Major players in social networking, such as Facebook, MySpace and LinkedIn, have already deployed mobile versions of their applications.

Moreover, mobile applications can be extended to support physical presence detection and thus eventually create a link and some kind of convergence between the virtual and real world. For example, Facebook mobile app lets you see which of your friends are around, while FourSquare allows users share their location with contacts, with a points-based rewarding policy.

In this context, data is necessarily “people centric”, namely, they do not only refer to the environment, as in traditional sensor networks, but they also reflect personal aspects of people and their social positions and roles. As a consequence, as also observed by Alex Sandy Pentland in his keynote at Sensys 2010[100], the main obstacle to this vision is users’ privacy and “unfortunately, current privacy law does not do so much for us”. The long term goal is to grant the control of private data to individuals instead of private companies as happens nowadays. In the future, a user should be able to delete her profile from a service provider or even to force the same service provider to pass her profile to a competitor before deleting it. However, we are still far from achieving this goal. In current systems the enforcement of privacy policies is mainly based on agreements between users and their service providers; users rely on their service providers and their technological infrastructures to securely manage their personal data.

On the other hand, while western countries are experiencing the increasing availability of high speed connections and the diffusion of last generation smartphones with advanced interfaces to access mobile social networks, many still consider Short Messages the most convenient means for instant message exchange 1. In any case, SMS traffic is still a consistent part of non-voice traf-

---

1 “If you look at instant messaging, e-mail or even social networking, they don’t have the ubiquity and the reach to replace messaging” - Bill Dudley, Sybase 365’s group director for
fic. According to Lloyd’s [7], overall Person-to-Person SMS traffic has been 4.5 trillion of messages in 2008. These figures seem to justify the investments of some companies in social networking applications based on Short Messages, such as Jyngle2 [4] and Peekamo [5]. Furthermore, in large parts of the world, in particular Asia and Africa, SMS are expected to remain the primary means for data communication, at least in the near future. In 2007, nearly 1.5 trillion mobile messages were sent in the Asia-Pacific region [41].

Mobile social networks are thus raising in popularity, but along with clear benefits for users and companies, some concerns primarily related to privacy issues are arising. In the last W3C Workshop on the Future of Social Networking [9], several position papers on this issue appeared. For example, the basic operation of establishing a “friendship” in a social network, whatever the term means for the specific application, is a simple operation (e.g. a mouse-click), but it necessarily entails trust in the likely exchange of private information. As a matter of fact, privacy is one of the main concerns in mobile communities. As Jeff Chester, executive director of the Center for Digital Democracy, put it: “The fact of the matter is that the business model they have developed for mobile advertising is one where lots of user data is collected and user profiles are analyzed” and “You’re talking about multiple layers of surveillance at the heart of the mobile marketing business model that raise serious privacy concerns.”

In this chapter we describe an application based on Short Messages (SMs) and local information available on mobile phones to realize a fully decentralized application for recommending new contacts in the social network of mobile phone users. Recommending new contacts is a basic service provided by virtually every social network application. With respect to existing solutions, our approach is characterized by some distinguishing features:

The social networking application we propose is completely decentralized. This implies that the social network is not maintained in a centralized fashion, as usually done in nowadays social networking applications, but it is updated and managed in a fully distributed way by the collective effort of user devices. In the scenario we envision, the application (that might for instance be down-

---

2Jyngle closed in August 2009.
4.1. MOBILE SOCIAL NETWORKING

loaded as an app) runs on commercial devices and it transparently collects and processes user information that is accessible in any mobile phone, such as the log of calls, the list of contacts or the inbox/outbox of short messages, possibly enriched by user profile information. This information is used to recommend new contacts.

The techniques we propose greatly reduce the amount of personal information that is disclosed, since it is exchanged with other users in the form of a compact summary that allows limited extraction of private data.

Information necessary to implement the application is exchanged transparently and opportunistically, by using the residual space in standard short messages occasionally exchanged by users. As a consequence, we do not ask users to change their habits in using SMS.

We want to emphasize that, though for simplicity we stick to a specific case study in which a user profile is minimalistically extracted from the log of calls, the general approach we propose is agnostic with respect to i) the information that defines the profile and ii) the technique we use to represent it in compact form. As to point i) in particular, we have only considered information that is always available in nowadays commercial devices, but it is clear that the profile of a user can be enriched by many other features, depending on their availability and the user’s privacy concerns.

The rest of this chapter is organized as follows: we discuss related work in Section 4.2. In Section 4.3 we describe the approach we follow. In particular, we discuss some social networks naturally arising when analyzing the behaviour of users in a (mobile) telephone network. We then discuss the issues arising in the recommendation of new contacts in such networks, in the first place the notion of similarity between users. In Section 4.4 we review and discuss the application of sophisticated hashing techniques that allow to estimate the degree of similarity between users in a fully decentralized and privacy preserving way. In Section 4.5 we discuss experimental work assessing the effectiveness of our approach on real, publicly available datasets. Finally, in Section 4.6 we summarize the main findings of this work.
4.2 Related Work

Past research has also considered decentralized systems for item recommendation, such as in [92], where the authors propose the P2P-based PocketLens architecture. We are aware of this body of work, but recommending items using statistical information about past user transactions is not the focus of this work, which is rather on the related but well distinguished “social matching” problem [121], in which we want to infer the latent structure of a social network.

This work has been proposed in [18] in which we provided some preliminary experimental results on a relatively small dataset. The experimental analysis here discussed is more exhaustive and also considers a much larger dataset.

Following the approach of [18], namely employing Jaccard coefficient and MinHash (i.e. sketches), Blundo et al. [27] presented EsPRESSo, a practical construction for privacy-preserving evaluation of sample set similarity. In that paper, privacy is enforced using PSI-CA [43]. As a consequence, both computation and communication costs are reduced, but the proposed protocol requires the Random Oracle Model under the One-More-DH assumption [21], whereas the protocol proposed in [18] is secure under the Decisional Composite Residuosity assumption [97].

The paper of Blundo et al. has also the merit of clearly showing the applicability of privacy preserving set similarity to several application domains such as, document similarity [95], biometric identification [20, 26, 106] and multimedia file similarity [51].

4.3 Social networking over SMS messaging

In this work, a node in the social network of mobile phone users is a mobile phone subscriber generating some amount of user-to-user communication. A link connecting two nodes, represents an ongoing social relationship (e.g. nodes are friends, colleagues, classmates, etc.) between the corresponding users. In our approach, this social relationship can only be inferred estimating the users’ social profiles similarity. Speaking in general terms, two users are similar when their social profiles are similar. In fact, the profile of a user is a general notion that depends on the information available to the system. In some cases
this includes some biographical data, such as date of birth, sex, information about tastes, interests or activities. A profile is also completed by information that can be extracted transparently from the system, without explicit user intervention, such as the log of calls, the list of contacts or the inbox/outbox of short messages.

We stress that in many cases, even limited information, e.g., example the address book or the log of calls, can be used to infer possible relationships: for example, two users appearing in each other’s address books are likely to be socially related, be it through a shared interest, a professional relationship, or simply because they are friends.

Mining the social network underlying telephone traffic has been considered in the past, for example in [14, 15]. Here, there is a (possibly labeled) link from A to B if A calls B at some point. The main goal in [14, 15] was to study the way in which such networks evolve over time, so as to infer and analyze probabilistic generative models [96] describing their evolution.

In the following Subsections 4.3.1 and 4.3.2 we provide a high-level overview of i) how we recommend new contacts and ii) how we estimate user similarity, while the specific implementation of our approach in the scenario we consider in this paper is described in Section 4.4.

4.3.1 Recommending Social Relationships

Recommending new social relationships is one of the most basic services provided by social network applications. In our context, we are interested in strategies to recommend new contacts of potential interest to users. The challenge here is clearly to find contacts that are likely to share some common traits or, put differently, that are in some way “similar” to the user to whom the recommendation is being provided. As stated, this problem is very close to the link prediction problem studied by Liben-Nowell and Kleinberg [82], whose focus is on statistical indicators of social closeness and not on their efficient and decentralized computation.

More formally, if a node A recommends a node B to a third node C, A is suggesting a potential interest or utility for C in establishing a contact with B (unless this contact already exists). Recommendation is performed on the basis of knowledge about the social profiles $L(B)$ and $L(C)$, which are
used to estimate the extent to which $B$ and $C$ are “similar”. The underlying assumption, made more precise in Subsection 4.3.2, is that the more similar $B$ and $C$, the more likely it is that they either have a contact, or they might benefit from establishing one.

![Diagram of the process](image)

**Figure 4.1: A scenario**

Privacy requirements make the explicit exchange of private profile information or user contact lists unrealistic for applications. Furthermore, due to the opportunistic nature of the application we envision, data must fit into the residual space of person-to-person short messages and thus they must be represented in a compact form (i.e. a sketch). Figure 4.1 outlines the general application scenario we consider. In step 1, users A and B compute the sketches $sk(L(A))$ and $sk(L(B))$ of their respective social profiles. As observed before, a sketch is a compact representation of a user’s social profile preserving
her privacy. In step 2 and 3, A and B occasionally send a short message to C. The message space is partially filled with some personal text (e.g. SMS Text = “shall we meet this evening?”) while the residual space is exploited to deliver the sketches. Observe that users interact with the SMS as usual, while the residual space is transparently managed by a suitable application. In step 4, user C (i.e. the recommender) infers a high degree of similarity between A and B on the basis of their respective sketches. In steps 5 and 6, C eventually recommends a possible friendship to users A and B.

4.3.2 Locally inferring community structure

One of the main issues in recommendation systems for social networking is predicting the potential benefit of new links between users. In the fully decentralized scenario we consider here, this amounts to answering the following question: when should a user A recommend a contact between two other users B and C she is aware of? This in turn implies a number of other issues: i) What information about B and C does A combine in order to decide whether or not she should suggest a contact between B and C if not existing already; ii) how is this information obtained, manipulated and exchanged; iii) how are computational, storage and communication constraints met; iv) how is privacy preserved.

Alike many networking applications, we recommend new contacts on the basis of similarities between users. Thus, A will recommend B and C to establish a contact if A assesses that B and C are “similar”. In particular, if we view profiles as feature sets, we say that two users A and B are similar when their social profiles \( L(A) \) and \( L(B) \) overlap significantly. In this perspective, we estimate user similarity by the Jaccard coefficient \( J(L(A), L(B)) = \frac{|L(A) \cap L(B)|}{|L(A) \cup L(B)|} \), a widely accepted measure of similarity between sets. In the social networking scenario we consider, it captures the well known fact [82, 96] that social networks are densely connected at a local level or, roughly put, the folklore that two friends of the same person are significantly more likely to be friends than any two randomly chosen people. In the scenario we consider, the profile of a user is the set of her contacts, but it is clear that broader definitions are possible, in which user profiles are enriched with more, possibly heterogeneous features. The minimal choice we do in this work on one hand corresponds to
information that is virtually available in any current commercial device and that can be transparently retrieved. On the other hand, our results show that this feature alone allows to extract valuable information about the underlying social network.

4.4 Efficiently mining the social network of SMS users

A key aspect in the applications we consider is estimating the size of the intersection between the social profiles of two users in a fully decentralized way. More precisely, if a user C receives short messages from A and B, she should be able to estimate $J(L(A), L(B))$ from summary information about $L(A)$ and $L(B)$ piggybacked in the messages themselves. It is clear that short message size poses stringent constraints on the amount of information that can be piggybacked. This is at most 140 bytes, but recalling that we only use the residual space on the message, a variable number of those bytes will be occupied by the message body itself. We show below how to address these issues in the following way: i) we adapt a technique initially conceived for Web page similarity estimation to the scenario we consider. The adoption of this technique allows to compute compact summaries or sketches of each social profile, which in turn allow efficient estimation of the Jaccard coefficient between social profiles. The space required by the proposed sketches is in the order of a few tenths of bytes; ii) we address the issue of variable SMS size under the assumption [130] that SMS sizes are (approximately) uniformly distributed. Specifically, for those messages created in person-to-person communications, the length seems to evenly span the whole range of the allowed message sizes [130], whose maximum value depends on the encoding that is used for each message, but it is typically 140 bytes. In the following, we refer to profiles based on users’ contact lists, since contact information is locally available on virtually every recent commercial device. For this reason, the terms social profile and contact list will be used interchangeably. We emphasize that the techniques described in the remainder can be extended to more general notions of user profile.
4.4. EFFICIENTLY MINING THE SOCIAL NETWORK OF SMS USERS

4.4.1 Background: estimating Jaccard coefficient

Consider the set of possible contact identifiers. Recall that, as motivated further in this section, they can be regarded as integer numbers falling in the range $[n] = \{0, \ldots, n-1\}$ for suitable $n$. The only assumption we need is that they are unique, a constraint that is met in practice in the applications we consider, where they are users’ telephone numbers or a suitable representation of them. As a consequence, considered any two users A and B, their contact lists $L(A)$ and $L(B)$ may be simply regarded as two subsets of $[n]$. Our goal is to measure their overlap using the Jaccard coefficient: 

$$J(L(A), L(B)) = \frac{|L(A) \cap L(B)|}{|L(A) \cup L(B)|}.$$ 

A very simple and elegant technique to estimate the Jaccard coefficient has been proposed in several equivalent forms by Broder et al. [30, 31]. Assume we are able to choose a permutation $\pi(\cdot)$ mapping $[n]$ onto itself uniformly at random. For every $X \subseteq [n]$, denote by $\pi(X)$ the set of the images of elements in $X$ when $\pi(\cdot)$ is applied and let $\min(\pi(X))$ denote their minimum. Then it can be shown [30] that (i) considered a set $S \subseteq [n]$ and for every $a \in S$, $P[a = \arg \min(\pi(S))] = 1/|S|$; (ii) for every $S_1, S_2 \subseteq [n]$: $P[\min(\pi(S_1)) = \min(\pi(S_2))] = J(S_1, S_2)$. This property immediately yields a technique to estimate $J(S_1, S_2)$.

The algorithm consists in performing $m$ independent executions of the following procedure: i) pick one permutation $\pi(\cdot)$ of $[n]$ uniformly at random from the $n!$ possible ones; ii) in the $i$-th iteration, let $\min(S_1) = \min(\pi(S_1))$ and $\min(S_2) = \min(\pi(S_2))$. We increment a counter $C_m$ whenever $\min(S_1) = \min(S_2)$. At the end of the process, our estimation of $J(S_1, S_2)$ is $C_m/m$. Standard tools from probability theory tell us that $C_m$ is an increasingly (with $m$) accurate estimation of $J(S_1, S_2)$.

4.4.2 Computing and maintaining contact list sketches

Unfortunately, generating permutations uniformly at random requires a number of truly random bits that is in the order of $n$ [30]. Fortunately, suitable families of simple, linear hash functions perform well in practice (e.g. see [28, 73]). In particular, we use linear permutations [28], i.e., functions of the form $h(x) = ((ax + b) \mod p) \mod n$. Here, $p$ is large prime, while $a$ and $b$ are integers belonging to the intervals $[1, p - 1]$ and $[0, p - 1]$ respectively.
CHAPTER 4. AN OPPORTUNISTIC RECOMMENDATION SYSTEM FOR MOBILE RESOURCE CONstrained DEVICES: CASE STUDY

UPDATE(Sk(A), pn)

Require: Sketch Sk(A), number pn
1: x = hash(pn) \{Hash pn to an integer in [n]\}
2: for i: 1 \ldots m do
3:    M_i = h_i(x) \{Map x according to a random permutation\}
4:    if M_i < min_i(A) then
5:        min_i(A) = M_i
6: end if
7: end for
8: return Sk(A)

Figure 4.2: Update algorithm.

We next describe how each node A of the network maintains the local sketch Sk(A) associated to L(A). As pointed out before, we assume below that every number in L(A) is an integer falling in [n]. To this purpose, it is enough to perform a first step in which each contact identifier (e.g., a user’s mobile phone number) is regarded as a string and this string is mapped onto an integer in [n], using any hash function, as long as the probability of collision is sufficiently small. This is for instance the case if we hash contact identifiers to 32-bit integers using a good hash function, e.g., implemented in Java standard classes. As a second step, \(m\) hash functions are generated. The \(i\)-th hash function has the form \(h_i(x) = ((a_i x + b_i) \mod p) \mod n\). The integers \(\{a_1, b_1, \ldots, a_m, b_m\}\) are generated independently and uniformly at random, respectively in the interval \([1, p-1]\) for the \(a_i\)’s and \([0, p-1]\) for the \(b_i\)’s. Finally, for \(i = 1, \ldots, m\), let \(\text{min}_i(A) = \min_{x \in L(A)} \{h_i(x)\}\). The sketch of \(L(A)\) is the ordered vector \(\text{Sk}(A) = (\text{min}_1(A), \ldots, \text{min}_m(A))\). A version of this algorithm that allows dynamic updates when new numbers are added to the contact list is given in Figure 4.2.

The cost of algorithm UPDATE(Sk(A), pn) is \(O(m)\). The deletion of items from the contact list is more expensive, since the element removed might be the one achieving minimum value on one or more of the hash functions. Therefore, in the case of deletions Sk(A) has to be recomputed from scratch and the cost becomes \(O(m|L(A)|)\). Note however, that \(m\) is in the order of a few tenths at most (with 10 already providing satisfactory results in the experiments). This complexity is therefore fully compatible with standard commercial mobile
4.4. EFFICIENTLY MINING THE SOCIAL NETWORK OF SMS USERS

phones.

In addition to $\text{Sk}(A)$, A’s device stores $\text{Sk}(B)$, if available, for every B in her contact list. The required amount of additional memory, as discussed further in greater detail, is a few tenths of bytes for each entry in the contact list (40 in case 10 hash functions are used and we use 32-bit unsigned integers), thus perfectly compatible with standard commercial devices.

4.4.3 Exchanging sketches

In the scenario we envision, if both users A and B run the application and B sends an SMS to A, B will use the available free space of the message to send its own sketch $\text{Sk}(B)$, or part of it, to A. Let’s assume for the moment that there is enough residual space in the message to send the whole $\text{Sk}(B)$. Whenever A’s device receives the message, it transparently extracts $sk(B)$ from the message body. If B is one of A’s contacts, then $\text{Sk}(B)$ is stored in A’s contact list, associated to B, possibly replacing an older copy of $\text{Sk}(B)$. Note that this is likely to be often the case since, as we see later, the size of a sketch is typically a few tenths of bytes, 40 in the present implementation. Moreover, we discuss how to address cases in which the SMS free space is not sufficient to contain $\text{Sk}(B)$ in a further paragraph of this section.

4.4.4 Fully decentralized recommendation of contacts

Recall that we assume that two users are similar to the purpose of the application whenever their contact lists overlap significantly. The algorithm in Figure 4.3 implements this general idea. In particular, the algorithm describes the behaviour of the generic, mobile terminal of some user A. If A has the sketches of both B’s and C’s contact lists, A will recommend B (C) to C (B) whenever the local estimation of $J(L(B), L(C))$ exceeds some given threshold $\theta$. In Section 4.5 we study, among others, how the choice of the threshold affects the quality of recommended contacts.

4.4.5 Implementation issues

If we consider the generic node A, the amount of memory needed to store its contact list is $\Theta(L(A))$. In our implementation, A also needs to store i) its
RECOMMEND(A, Sk(B), Sk(C), θ)

Require: Node A, Sketch Sk(A), Sk(C), threshold θ

1: Estimate \( J(L(B), L(C)) \) from Sk(B) and Sk(C) \{Node must have both\}
2: if \( J(L(B), L(C)) > \theta \) then
3: A recommends B to C or vice versa
4: end if

Figure 4.3: Recommendation algorithm.

own sketch Sk(A) and, in the worst case, ii) Sk(B), for a subset of nodes from which A received SMS messages in the past. If we assume that A stores the sketch of every contact, the required amount of memory is \( O(m(|L(A)|)) \). In practice, if we use \( m = 10 \), the additive amount of bytes required for each contact is about 40. This is in the same order of magnitude of an entry in any address book of a commercial device.

The computational cost of maintaining sketches and providing recommendations is also compatible with current commercial devices. In particular, adding a new contact to the contact list of a node A requires updating Sk(A) (algorithm UPDATE(· · ·) in Figure 4.2) and has cost \( O(m) \). Removing a contact from L(A) (typically a less frequent operation) is more expensive but it has (up to \( m \)) still linear cost, i.e., \( O(m|L(A)|) \). Finally, for two nodes B and C other than A, deciding at A as to whether recommending each of them to the other requires estimating \( J(L(B), L(C)) \), which has cost \( O(m) \). Computation is performed at user devices. Nowadays, these are typically small computers, whose capabilities are perfectly compatible with the computational effort required by the proposed techniques.

The number of hash functions required (i.e., \( m \)) is chosen, so that probability that the estimation of the Jaccard coefficient differs from the true value by more than a chosen constant is below a suitably small constant. We refer the reader to specific work (e.g., [30, 31]) for technical details. In our case, experimental evidence suggests that already 10 hash functions are sufficient to strike a reasonable balance between accuracy of the estimation and memory requirements.

A further constraint is that all user devices use the same set of hash functions. In practice, hash functions and the algorithms we propose will be imple-
mented and maintained in the device’s memory. This in turn requires storing, for each hash function, its coefficients and \( p \) in binary form. In our implementation, coefficients are 32-bit integers, while \( p \) is the well-known Mersenne prime \( 2^{31} - 1 \), which does not need to be stored explicitly. So, it turns out that the actual storage requirements for maintaining hash functions is around 80 bytes. The overall implementation (code, hash functions, runtime data structures) requires less than 1Kbyte space. To this, we must add the (variable) size of the user’s (modified) contact list. Thus, the storage requirement of the modified contact list is in the same order of magnitude as in a standard implementation.

We observed earlier that we cannot always assume that the message body of an SMS sent from some node A to another node B has enough free space to host \( \text{Sk}(A) \). The most direct way to circumvent this problem is for A to send its sketch whenever the available free space in the message body exceeds \( |\text{Sk}(A)| \). In fact, the distribution of SMS message sizes seems to be approximately uniform [130]. Assuming for the sake of simplicity that it is exactly uniform and that message sizes of different messages are independent variables, we have that half of the messages have 80 bytes available space in the average, more than 75\% have at least 40 bytes available to carry sketches and so on. This means that, in the average, 1.34 message are enough for A to send its sketch to B, which means that, in practice, if A sends 2 SMS to B, the latter is very likely to receive A’s sketch. \(^3\)

4.5 Experimental Analysis

In this section we present the results of experimental analysis on real, publicly available data sets, in our opinion supporting the effectiveness of the approach we propose.

\(^3\)An alternative solution is that A sends to B part of its sketch, compatibly with the available space in the SMS message body. This solution requires keeping both at A and B, to keep track of the portions of \( \text{Sk}(A) \) still missing at B. In fact, the former solution can be more easily implemented than the latter and it requires no additional data structures.
4.5.1 Objectives

Our experimental work had the following main goals. In the first place, we wanted to understand the intrinsic effectiveness of the Jaccard coefficient to infer social relationships in the network of mobile phone users. A further issue was to assess whether the techniques we use to approximate the Jaccard coefficient and discussed in Section 4.4, are compatible with the hard space constraints, imposed by Short Message size. In particular, these severely limit the number of hash functions that can be used to compute contact list sketches, which in turn affects the accuracy of the estimation, especially when the value of the Jaccard coefficient is relatively small in absolute terms, as is the case for one of the datasets we consider. Finally, we wanted to investigate the effectiveness of our overall approach in suggesting contacts to users. The problem here is that the public datasets we used did not allow us to directly assess the \textit{a posteriori} effect of recommendations on users’ choices. For this reason, in our experiments we considered the ability of our approach to predict existing links as a proxy for its effectiveness in providing useful recommendations.

4.5.2 Data sets and contact graphs

In this subsection, we describe the datasets we used and how we extracted from them the social networks we considered for the experiments. The first dataset contains data obtained from a sample of real mobile phone users over a relatively large time interval. As such, the dataset is ideal to test the effectiveness of the recommendation strategy we propose. On the other hand, the sample size is about 100 users among which a few thousand contacts (i.e., phone calls) were recorded. Furthermore, the sample might be statistically biased, since the involved people belong to the same university campus. For this reason, we also performed our experiments on a far larger dataset. This refers to a different application (i.e., it is a sample of the social network of Facebook users), but some general structural properties of both networks (e.g., degree distribution and clustering coefficient) are similar and follow patterns largely observed in social networks, as further commented in the paragraphs that follow.
4.5. EXPERIMENTAL ANALYSIS

The Reality Mining Project dataset. Accessing telephone traffic data is far from trivial, since very few public datasets are available. The Reality Mining project [2, 49] represents the largest mobile phone experiment ever attempted in academia. Its dataset contains thousands hours of continuous data on daily human behavior and contains information on call logs, Bluetooth devices in proximity, cell tower IDs, application usage, phone status.

The Facebook dataset. In order to perform a more exhaustive and complete validation of the discussed techniques, we replicated the experiments on a much larger, real dataset obtained from [8]. This dataset is the result of a sampling based on the Metropolis-Hastings Random Walk (MHRW) [117, 118] over the graph of Facebook users. This technique yields a uniform stationary distribution of nodes (users), resulting in a dataset with 957K unique nodes and their relationships.

Contact graphs. We extracted a contact graph, i.e., a graph describing contacts among users of the two applications, from the two datasets we considered. For the Reality Mining project dataset, we used call logs to build a contact graph where nodes are mobile users characterized by unique ids (i.e., telephone numbers) and there is an edge connecting two users $i$ and $j$ if and only if the call log contains at least 1 call occurring between $i$ and $j$. Formally, in this case the contact graph $G(V,E)$ was obtained as follows:

- $V$ is the set of users appearing in the log of calls

- for each pair $(i,j) \in V$, edge $(i,j) \in E$ if and only if at least 1 call occurred between $i$ and $j$.

To avoid problems related to data incompleteness, we restricted our experiments only to the people actually participating in the Reality Mining project (around 100 people), whose logs are complete and accurate.

For the Facebook dataset, we directly used friendship relationships to construct a contact graph. In particular, there is an (undirected) link between nodes $i$ and $j$ whenever the corresponding users are friends in the Facebook social network.
Note that the contact graphs for both cases are undirected. This is perfectly consistent with Facebook’s dataset, in which contacts represent friendships in the social network. For the Reality Mining Project dataset, this entails assuming that contact lists are symmetric, i.e., $i$ belongs to the contact list of $j$ (and vice versa) if at least 1 call occurred between $i$ and $j$ during the period of observation.

**Comparison between datasets.** Even though the application scenarios for the two datasets we consider are different, some important statistical properties of the resulting social networks are similar. For example, the degree distributions for the two networks both follow power laws and their upper parts are roughly approximated by a Zipf law with parameter close to 1. Also, the clustering coefficient of the Facebook contact graph is 0.0021. This is the same order of magnitude as the Reality Mining Project contact graph (0.0095), even though significantly smaller. This may be because the former dataset refers to a statistically biased, relatively small sample of people who are more likely to know each other. Finally, as we show in Subsection 4.5.5, both datasets show the same, strong correlation between overlap in contacts lists and social ties. This similarity is not surprising, but it is rather a well known common trait to many social networks describing human interactions [96].

**Remark.** For the Reality Mining Project dataset it is possible to define different contact graphs, describing networks of increasingly strong social ties. In particular, it is possible to define a graph $G_w(V, E)$ in which we declare the existence of a link between $i$ and $j$ if and only if at least $w$ calls occurred between them. In this way, we can filter out occasional contacts. Though this is potentially interesting, in our experiments we restricted to $G_1(V, E)$, since the Reality Mining Project dataset is relatively small and higher values of $w$ further reduced its size.

### 4.5.3 Experimental scenario for recommendations

We assessed the quality of our technique in providing recommendations of good quality by using its ability to uncover existing relationships as a proxy. In particular, for each node $v_i$ and for each pair $\{v_a, v_b\}$ both belonging to $v_i$’s contact list, we ran our algorithm to predict the existence or non-existence
4.5. EXPERIMENTAL ANALYSIS

Require: $G_w(V, E)$
1: for $v_1 \ldots v_n \in V$ do
2: for each $v_a, v_b$ among the contacts of $v_i$ do
3: retrieve $Sk(A)$, $Sk(B)$ by emulating an SMS reception
4: RECOMMEND($A$, $Sk(A)$, $Sk(B)$, $\theta$)
5: end for
6: end for

Figure 4.4: Simulation algorithm.

of link $(v_a, v_b)$. We then checked whether the link existed or not. This is synthetically described by the algorithm in Figure 4.4, which simulates the general recommendation algorithm described in Section 4.4. As to user profiles over which sketches are computed, for each user $v$, we considered the set of $v$'s neighbours as her social profile, namely her list of contacts. As remarked earlier, this is a minimalistic assumption and corresponds to information that can be transparently accessed on virtually any current commercial device.

4.5.4 Performance indices

The error of the recommendation strategy we propose is potentially affected by two factors: i) inaccuracy in the estimation of the actual value of the Jaccard coefficient; ii) error in the recommendation itself, i.e., the contact we recommend is not interesting to the user. These two aspects are clearly interrelated in complex ways. We considered these two contributions separately, which corresponds to the worst-case assumptions that the effects of the two sources of error sum up.

4.5.4.1 Effect of accuracy in the estimation of the Jaccard coefficient.

Assessing the accuracy of our approximation of the actual Jaccard coefficient poses some issues. In the first place, our data show that even values of the Jaccard coefficient related to a significant degree of social relationship can be low in absolute terms. This makes an accurate estimation harder to attain given the stringent constraints we have to comply with. In particular, if we use $m$ hash functions, we only have $m$ possible values for our estimation of
the Jaccard coefficient. When \( m = 10 \) as we assume, this provides very little granularity. Namely, possible values of the estimated Jaccard coefficient are \( 0.1j \), with \( j = 0, \ldots, 10 \), whereas values of the true Jaccard coefficient corresponding to a significant degree of social interaction are around \([0.05, 0.1]\) in the Reality Mining Project dataset.

On the other hand, our algorithm is threshold-based: it recommends a contact between two nodes \( A \) and \( B \) whenever \( J(L(A), L(B)) \) is above a given threshold. For this reasons, we consider the \emph{Jaccard-Estimation Performance (JEP)} index, defined as the fraction of times that our algorithm gives the same recommendation as it would give if it knew the exact values of the Jaccard coefficient. We call these two versions of the algorithm \emph{apxJacc} and \emph{exactJacc} in the paragraphs that follow. Formally, for every node \( i \), let \( C_i \) denote the number of times that \emph{apxJacc} and \emph{exactJacc} take the same recommendation decision for pairs of nodes belonging to \( i \)'s contact list.

The \emph{Jaccard-Estimation Performance (JEP)} index is formally defined as:

\[
JEP = \frac{\sum_{1 \leq i \leq |V|} C_i}{t}
\]

where \( V \) is the vertex set, i.e., the overall number of users and \( t \) is the overall number of node pairs evaluated.

### 4.5.4.2 Quality of recommendations.

As observed earlier, the datasets we used do not allow to test the a posteriori effect of recommendations. For this reason, we tested the ability of our approach to provide high quality recommendations to users as a proxy for its effectiveness. To this purpose, we checked to which extent the contacts that are recommended correspond to actual links, thus evaluating \emph{precision} (i.e., the fraction of existing links that have been recommended over the total number of given recommendations) and \emph{recall} (i.e., fraction of all existing links that have been recommended over the total number of actual links) of our recommendation algorithm.
4.5. EXPERIMENTAL ANALYSIS

4.5.5 Experimental results

In this section, we provide experimental results that address the following issues: i) whether or not the Jaccard coefficient is a good indicator of social ties in the datasets we consider and which are reasonable threshold values for the recommendation heuristic we propose; ii) how good is our estimation of the Jaccard coefficient, at least in the sense made precise in the previous subsection; iii) how good are the recommendations we provide, which we indirectly answer by the extent to which we are able to infer existing contacts between node pairs.

4.5.5.1 Experimental setting

The algorithms we propose depend on the similarity threshold $\theta$ and on the number $m$ of hash functions used to generate sketches of user profiles. Furthermore, they are probabilistic in the generation of the hash functions used to estimate the Jaccard coefficient (see Subsection 4.4.2). For this reason, to generate the results described in the following Subsections 4.5.5.3 and 4.5.5.4, we performed 10 independent runs of the algorithms for every pair of $(\theta, m)$ we considered and we averaged the results over the 10 trials. In particular, in each run we generated $m$ hash functions, independently at random according to the guidelines given in Subsection 4.4.2. We used Java Math’s built-in pseudo-random generator to produce the coefficients of each hash function, each time initializing the pseudo-random generator object with a different seed to enforce independence.

As to the values of $\theta$ and $m$, we considered $\theta \in \{0.0, 0.01, 0.02, ..., 1.0\}$, while for the number of hash functions we took $m \in \{10, 20\}$ for the Reality Mining Project dataset and $m \in \{10, 20, 40, 60, 80, 100\}$ for the Facebook dataset. The reason for considering a larger range of values for $m$ in the Facebook scenario was testing the effect of increasing the number of hash functions beyond values that are of practical interest in the scenario we envision but might be feasible in others.
4.5.5.2 Jaccard coefficient and social ties

Figure 4.5 synthetically describes the correlation existing between values of the Jaccard coefficient and existence of links between node pairs for the Reality Mining Project dataset. More in detail, the $x$-axis is divided into intervals of width 0.05 each, starting at 0.0 and ending at 0.2. For the $j$-th interval ($j = 0, 1, 2, 3$), the ordinate represents the fraction of pairs $(A, B)$ of users such that i) $J(L(A), L(B))$ falls in the interval $[0.05j, 0.05(j + 1)]$ and ii) $A$ and $B$ are contacts, i.e., they are in each other’s contact lists. The $x$-interval stops at the value 0.2, since we observed too few pairs with Jaccard coefficient beyond this interval, to be statistically meaningful. This picture clearly shows that the Jaccard coefficient is a good indicator of social ties in mobile user networks. Furthermore, at least in the datasets we considered, the Jaccard coefficient allows to identify a sharp transition around the value 0.05, from a region characterized by sporadic ties to one characterized by frequent social relationships. In light of these observations, we chose the value 0.05 as a threshold in our recommendation algorithm.

![Figure 4.5: Existing linked pairs over total pairs](image)

Figure 4.5 reports similar results for the Facebook dataset. The $x$ and $y$-axes have the same meaning as before, with the only difference that, given the much larger size of the dataset, in this case we have a statistically significant number of pairs with very high values of the Jaccard coefficient, so that this time the $x$-interval extends to the value 1. These results confirm the strong
4.5. EXPERIMENTAL ANALYSIS

positive correlation between values of the Jaccard coefficient between pairs’ contact lists and social ties. An interesting difference is that this time, the threshold in the Jaccard coefficient above which we have a significant amount of social interaction is higher than in the previous case and there is no sharp transition. This can probably be explained by the fact that it seems to be easier to offer “friendship” on a social network than one’s private phone number. Furthermore, on Facebook one can see (some of) the feeds of her friends’ friends, while it is relatively easy for two perfect strangers to share one common acquaintance. A further lesson is that the right choice of the threshold is to some extent application specific, even though typical values seem to lie between 0.1 and 0.3 (see also results in Subsection 4.5.5.4).

![Figure 4.6: Existing linked pairs over total pairs](image)

4.5.5.3 Jaccard estimation performance

**Reality Mining Project dataset.** Figure 4.7 shows the behaviour of the Jaccard-Estimation Performance, as defined in the previous subsection, as a function of the threshold, both when 10 and 20 hash functions are used to estimate the Jaccard coefficient. In particular, the function has been computed in 5 points for 20 hash functions. Recall that each point represents the average taken over 10 independent runs of the algorithm.

More precisely, for \( j = 1, \ldots, 5 \), the runs of the \( j \)-th batch were executed
with threshold value $0.05j$. For 10 hash functions we only report two points, since for values of the threshold above 0.1 the distance between the two curves becomes smaller and smaller. Results show that the algorithm that estimates the Jaccard coefficient using hash functions takes the same decisions as the one using the exact value of the Jaccard coefficient in the vast majority of cases. This means that, even under the stringent constraints for sketch sizes, we are able to follow the ideal algorithm pretty close, as far as the recommendation decision is concerned.

![Figure 4.7: Jaccard-Estimation Performance for Reality Mining Project dataset.](image)

**Facebook data set.** Figure 4.8 summarizes results for the Jaccard Estimation Performance index for the Facebook dataset. In this case, we considered a larger range for the number of hash functions, in order to also test the effect of increasing this number beyond values that are of practical interest in the SMS scenario but might be feasible in others. The general trend is clear: similarly to the previous case, for values of the threshold beyond 0.1, the hash-function based and the true Jaccard-based heuristic essentially provide the same recommendations. Two interesting features of these results are i) the slight decrease in JEP when the threshold grows from 0.8 to 0.9 and ii) the decay in the JEP
for the curve corresponding to 10 hash functions as the threshold grows from 0.1 to 0.2, which are discussed below.

Two opposite mechanisms are at work, that help explain the behaviour of the curves in Figure 4.8: on the one hand, as threshold grows, all heuristics are going to recommend new pairs when the intersection of the respective contact lists is higher in terms of Jaccard coefficient. On the other hand, as the latter grows, the hash function-based and the Jaccard-based heuristics are more likely to provide the same recommendations. At the same time, as threshold grows, both the hash-based and the Jaccard-based heuristics provide less recommendations overall. As a result, the marginal impact of cases where the two heuristics take different decisions increases. This latter effect explains the slight decay in the JEP curve when the threshold grows from 0.8 to 0.9. As for the decay in the JEP for the curve corresponding to 10 hash functions as the threshold grows from 0.1 to 0.2, the data show that the number of recommendations given by the Jaccard-based heuristic decreases much more than for the 10 hash function based heuristic in this interval, whereas the number of failures for the latter does not decrease as much. The reason for this fact is that for a threshold 0.2 the inaccuracy in the estimation of the Jaccard coefficient is still relatively high when using only 10 hash functions.

Figure 4.8: Jaccard Estimation Performance for Facebook dataset.
4.5.5.4 Quality of recommendations

Reality Mining Project dataset. Figure 4.9 shows the effectiveness of our algorithms in predicting the existence of contacts in the social network of mobile users. In particular, Figure 4.9 is a scatter-plot showing the trade-off between precision and recall as the threshold and number of hash functions used vary. For a better reading, values with precision $< 0.2$ or recall $< 0.2$ have been filtered out. The following remarks are in order: i) the best trade-off between precision and recall is struck near the interval $[0.05, 0.1]$ of the threshold, both for the algorithm using exact Jaccard coefficient and for our heuristics; ii) For higher values precision increases and recall decreases, meaning that on one hand, a similarity beyond the threshold implies a contact with increasing probability, but we omit to recommend many contacts that fall below the threshold; iii) the values of precision/recall we obtain for the best choice of the threshold fall in the interval $[0.4, 0.6]$. Such values are indeed relatively high, since they refer to the prediction of really existing links; if we were only recommending links that already exist, there would be no point in providing recommendations.

Facebook dataset. Figure 4.10 shows that the trend emerging in the relatively small Reality Mining Project dataset is confirmed in the much larger Facebook one. Again, Figure 4.10 is a scatter-plot showing the trade-off between precision and recall as the threshold and number of hash functions used vary. For the number of hash functions, we only report results for the cases $m = 10, 40, 100$ for the sake of readability. In the picture, $\theta$ is again the value of the threshold, with $\theta \in \{0.1, 0.2, \ldots, 0.9\}$ and increasing as we move from left to right. Then, for a given value of $m$ or assuming the exact Jaccard coefficient is used to measure similarity, the behaviour is clear and intuitive as in the previous case: an increase in the threshold will improve precision (a link between two nodes is assumed when the overlap in their contact lists is higher) at the expense of recall (we are more likely to neglect links between pairs whose contact lists overlap below the threshold). In this perspective, in Figure 4.10, we emphasized three values of the threshold $\theta = 0.1, 0.3$ and

---

4 The threshold is represented as a label over each point in the scatterplot.

5 This is the reason why only one point of the 10 hash algorithm is represented on the scatterplot.
0.9 that are in some way representative of the different patterns arising in the results as the threshold varies. In particular, dashed ellipses group points that correspond to results for the same value of the threshold. If we do this for all values of the threshold, we note that points corresponding to the same value of the threshold tend to be grouped together, meaning that different heuristics perform similarly for the same value of threshold. In particular, this is always true when 20, 40 or 100 hash functions or the exact value of the Jaccard is used to measure similarity. Finally, we note that, even with 10 hash functions, setting the threshold to 0.3 allows to strike a very good balance between precision and recall for the Facebook dataset, with high values for both parameters and similar to those obtained for the Reality Mining Project dataset.

4.6 Final considerations

These results in our opinion provide an indication that our fully decentralized strategies might prove effective in providing recommendations of good quality. Furthermore, even though the Reality Mining Project dataset is relatively small and is likely to have some statistical bias, the application of these algorithms to a more realistic dataset such as a sampling of the Facebook graph,
confirms the goodness of the approach.

Moreover, our results indicate that estimating similarity of users’ contact lists already represents a powerful feature to provide useful contact recommendations, thus allowing the application to be compliant also with resource constrained mobile phones. Considering these results as a lower bound on performances, we expect that the application of it to modern smartphones, with enriched user profiles and resources can lead to even better performances.
Conclusions

In conclusion, this thesis addresses the emerging area of people-centric sensing, a research field that has gained the spotlight in the last years, and is likely to become much more important as technology and personal devices evolve towards a more and more pervasive approach. Although many consider PCS as a natural evolution of WSN, we have stressed that these networks are quite different from traditional wireless sensor networks as well as the challenges they offer. Literature in this field is still at an early stage, and many works are mainly focused on architectures, frameworks and real world applications, while the subject of users’ privacy still lacks of tangible solutions. Under these circumstances this thesis aims at raising important questions about privacy in people-centric networks and discusses a number of mechanisms and protocols that can apply to a wide range of applications to provide the desired confidentiality of users’ personal information.

We have started discussing the evolution of technology and approaches that brought, in the last decade, from wireless sensor networks to the rise of people-centric sensing. In chapter 1 we have analyzed the state of the art, together with the main aspects involved in this process and the main research results that have faced the main challenges of this new approach. Moving into a case study, the first part of this thesis has been focused on the aspects related to environmental monitoring. In this common scenario, two works have been presented, the first based on an opportunistic people-centric approach, in which we have discussed a typical architecture, mostly common to the vast majority of PCS applications. Under this architecture, we have focused on privacy of georeferenced data since this is one of the main issues to deal with, when users’ personal information are involved in a sensing process. We provided techniques and algorithms to collect, share and elaborate data, without
explicitly disclosing private information, guaranteeing a suitable privacy level, but still allowing a trusted, or even untrusted central authority to gather statistical data from them. We stress that even though we have focused on georeferenced data, the algorithms we discussed are independent from it and can be applied to any kind of sensitive data. Furthermore we have shown how privacy is depends and has impact on other aspects such as performances and data quality. All these are aspects are tightly coupled and a suitable trade-off has to be found to better achieve the desired service level in a real case scenario. Keeping this in mind, we have defined some metrics and performed a vast analysis of our mechanisms and algorithms to better understand the effects of each parameter on the whole system performances.

Derived from a previous research experience on static wireless sensors networks, in chapter 3 we have discussed the feasibility for this technology to be applied for environmental monitoring in the same application scenario. Since challenges involved in the use of WSN mostly differ from those deriving from PCS, we have focused on MAC protocols analysis regarding energy consumption and network performances, using direct measurement techniques. This first part has shown, by means of experimental analysis, how issues like scalability, that strongly limits the adoption of WSN in very large scale environments, can be overtaken with a people centric sensing approach that leverages human mobility and human carried devices. Nevertheless, this introduces new challenges such as privacy preservation, that can be addressed with suitable techniques. In the second part of the thesis, we have focused on the new opportunity provided by a people-centric approach, in particular the ability for a system to sample non-physical dimensions such as social interaction. In chapter 4 we have presented and analyzed an opportunistic fully-distributed recommendation system for mobile resource constrained devices. We have modeled a suitable distributed architecture and evaluated the quality of presented algorithms to give recommendations of friendship by opportunistically leveraging the residual space in SMS communication. On the basis of the application requirements, we have adapted the privacy preserving sketching techniques already used in chapter 2 to cope with the new constraints imposed by the application, and provided an experimental analysis of the algorithms to show
their effectiveness over two major real datasets, the first deriving from a real world experiment[49], the second extracted from Facebook’s social graph[8].

End note

Based on the actual trends of both industrial and scientific research, it is foreseeable that the challenges discussed in this thesis will become more and more effective, together with others that will probably arise as technology improvement makes possible an increasing symbiotic relationship between humans and personal devices. This will definitely open a new stage in which traditional small-scale networks and application-specific systems considered so far will give way to huge heterogeneous sensing networks, potentially made of million devices, and able to collectively perform tasks that were unimaginable few years ago. While we are quite sure that sensing capabilities of these devices will probably be enriched by means of new physical or virtual sensors (e.g.: mood, thoughts...), in the near future research will probably focus on the definition of overarching frameworks or architectures for people centric networks that capture these concepts towards the realization of something close to the definition of “internet of things & people”.
Bibliography


123
based node selection for probabilistic coverage in people-centric sensing.
In Sensor, Mesh and Ad Hoc Communications and Networks (SECON),
2011 8th Annual IEEE Communications Society Conference on, pages
134–142, June 2011. 28

Probabilistic coverage methods in people-centric sensing. Journal of

Probabilistic methods for spatio-temporal coverage in people-centric
sensing. IPSJ SIG technical reports, 2010(3):1–8, 2010-08-26. 28

for massive graphs. In Proceedings of the thirty-second annual ACM

massive graphs. In Bob Werner, editor, Proceedings of the 42nd IEEE
97

79

[17] C. A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati,
and P. Samarati. Location privacy protection through obfuscation-based
techniques. 21, 23

[18] Elisa Baglioni, Luca Becchetti, Lorenzo Bergamini, Ugo Colesanti, Luca
Filipponi, Andrea Vitoletti, and Giuseppe Persiano. A lightweight privacy
preserving sms-based recommendation system for mobile users. In Proceedings of the fourth ACM conference on Recommender systems,

anonymous location queries in mobile environments with privacygrid. In


[24] Jan Beutel, Stephan Gruber, Stefanie Gubler, Andreas Hasler, Matthias Keller, Roman Lim, Lothar Thiele, Christian Tschudin, and Mustafa Yıldızcel. The permasense remote monitoring infrastructure. 79


[86] Hong Lu, Wei Pan, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. Soundsense: scalable sound sensing for people-centric applications on mobile phones. In *Proceedings of the 7th international
conference on Mobile systems, applications, and services, MobiSys ’09, pages 165–178, New York, NY, USA, 2009. ACM. 39


[98] Andrew Parker, Sasank Reddy, Thomas Schmid, Kevin Chang, Ganerwal Saurabh, Mani Srivastava, Mark Hansen, Jeff Burke, Deborah Estrin, Mark Allman, and Vern Paxson. Network system challenges in selective sharing and verification for personal, social, and urban-scale sensing applications. In *In HotNets*, 2006. 16


annual international conference on Mobile computing and networking, MobiCom ’11, pages 73–84, New York, NY, USA, 2011. ACM 92


[122] Carmela Troncoso, Kasteelpark Arenberg, and George Danezis. The bayesian traffic analysis of mix networks. 20


