CrowdSenSim: a Simulation Platform for Mobile Crowdsensing in Realistic Urban Environments

Claudio Fiandrino, *Student Member, IEEE,* Andrea Capponi, *Student Member, IEEE,* Giuseppe Cacciatore, Dzmitry Kliazovich, *Senior Member, IEEE,* Ulrich Sorger, Pascal Bouvry, *Member, IEEE,* Burak Kantarci, *Senior Member, IEEE,* Fabrizio Granelli, *Senior Member, IEEE,* and Stefano Giordano, *Senior Member, IEEE*

***Abstract*—Smart cities take advantage and exploit the most advanced information technologies to improve and add value to existing public services. The Internet of Things (IoT) paradigm makes the Internet more pervasive where objects equipped with computing, storage and sensing capabilities are interconnected with communication technologies. Because of the widespread diffusion of IoT devices, applying the IoT paradigm to smart cities is an excellent solution to build sustainable Information and Communication Technology (ICT) platforms. Having citizens involved in the process through mobile crowdsensing (MCS) techniques augments the capabilities of the platforms without additional costs. For proper operation, MCS systems require the contribution from a large number of participants. Simulations are therefore a candidate tool to assess the performance of MCS systems. In this paper, we propose and present the design of CrowdSenSim, a simulator for mobile crowdsensing. CrowdSenSim operates in realistic urban environments, which makes it an excellent tool for analysis of smart cities services. We demonstrate the effectiveness of CrowdSenSim for the most popular MCS sensing paradigms, participatory and opportunistic and we present its applicability to a popular community service in cities, namely smart public street lighting.**

***Index Terms*—Mobile crowdsensing, simulations, smart cities.**

I. Introduction

WORLD population living in cities has experienced an unprecedented growth over the past century. While only 10% of the population lived in cities during 1900, nowadays this percentage corresponds to 50% and is projected to further increase [1]. Sustainable development plays therefore a crucial role in city development. While only 2% of the world's surface is occupied by urban environments, cities contribute to 80% of global gas emission, 75% of global energy consumption [2] and 60% of residential water use [1].

C. Fiandrino is with Imdea Networks Institute, Leganés, Madrid, Spain. E-mail: {claudio.fiandrino}@imdea.org. Claudio developed the current work as a Ph.D. student at the University of Luxembourg.

C. Capponi, U. Sorger, and P. Bouvry are with the University of Luxembourg, Luxembourg. E-mail: {firstname.lastname}@uni.lu.

D. Kliazovich is with Examotive, Luxembourg. E-mail: kliazovich@ieee.org.

G. Cacciatore, and F. Granelli are with the University of Trento, Italy. E-mails: giuseppe.cacciatore@studenti.unitn.it, fabrizio.granelli@unitn.it.

B. Kantarci is with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada. E-mail: {burak.kantarci}@uOttawa.ca

S. Giordano is with the University of Pisa, Pisa, Italy. E-mail: stefano.giordano@iet.unipi.it.

Smart cities rely on Information and Communication Technology (ICT) solutions to improve citizens' quality of life [3], [4]. The application of the Internet of Things (IoT) paradigm to urban scenarios is of special interest to support the smart city vision [4]–[6]. IoT is envisioned as the candidate building block to develop sustainable ICT platforms. With IoT, everyday life objects are uniquely identifiable and "smart", i.e., they are equipped with computing, storage and sensing capabilities and can communicate one with each other and with the users to enable pervasive and ubiquitous computing [7]. Including citizens in the loop with crowdsensing approaches augments capabilities of existing infrastructures without introducing additional costs and has been proved to be a win-win strategy for smart city applications [8]–[10].

Mobile crowdsensing (MCS) has emerged in the recent years, becoming an appealing paradigm for sensing data [11]. In MCS, users contribute data generated from sensors embedded in mobile devices including smartphones, tablets and IoT devices like wearables. Accelerometer, gyroscope, magnetometer, GPS, microphone and camera are just a representative set of sensors which are nowadays employed to operate a number of applications in many domains, including, among the others, health care, environmental and traffic monitoring and management [12]. To illustrate, Google exploits crowdsourced information about smartphones locations to offer real-time view of congested traffic on roads. Moreover, Google has recently released Science Journal, a new application which permits to gather and visualize data coming from smartphone sensors [13].

The aggregated information acquired through MCS platforms is typically delivered to a collector in the cloud and consumed according to a Sensing as a Service (S^2aaS) model (see Fig. 1). S^2aaS makes available to the public data collected from sensors. Consequently, companies have no longer the need to acquire an infrastructure to perform a sensing campaign. IoT and MCS are key enablers in the S^2aaS model, which in turn is envisioned to play are indispensable role in smart cities. Efficiency of S^2aaS models is defined in terms of the revenues obtained selling data and the costs. The organizers of a sensing campaign, such as government agencies, academic institutions or business corporations, sustain costs to recruit and compensate the participants for their involvement [14]. Also the users sustain costs while contributing data. These costs are the energy spent from the batteries for sensing and reporting data and, eventually, the data subscription plan if

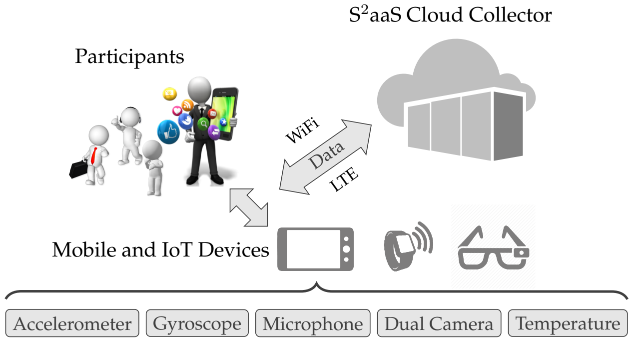


Fig. 1. Cloud-based MCS system

cellular connectivity is used for reporting.

In MCS, data acquisition or collection, can be *participatory* or *opportunistic* [12]. In opportunistic sensing systems, the user involvement is minimal: sensing decisions are application- or device-driven. In participatory sensing systems, users are actively engaged in the sensing process. The users, also called participants in the remainder of the paper, are recruited by a central platform, which dispatches sensing tasks. Users can then decide which request to accept and, after accepting, they have to accomplish specified sensing and data reporting tasks. On one side, opportunistic sensing lowers the burden of user participation as devices or applications are responsible to take sensing decisions. Conversely, participatory sensing systems are tailored to crowdsensing architectures with a "central platform", which facilitates system control operations like task assignment, user incentives and rewarding to compensate the participants for their contribution.

In this paper, we propose CrowdSenSim, a new tool for simulating mobile crowdsensing activities in realistic urban environments. CrowdSenSim is specifically designed to perform analysis in large scale environments for both participatory and opportunistic sensing paradigms. CrowdSenSim allows the researcher to investigate the performance of MCS systems, with a focus on data generation and participant recruitment, which is a pillar step in participatory sensing paradigms. The simulation platforms offers to the researcher the capability to visualize in an unprecedented fashion on urban maps the results on data generation. Moreover, it allows the researchers to assess at a fine-grained level the energy cost the participants spend both for sensing and reporting the produced data.

The contribution synopsis of this paper can be summarized as follows:

- Proposal of CrowdSenSim, a simulation platform for MCS systems deployed in realistic urban environments and presentation of its design features.
- Validation of CrowdSenSim's performance for opportunistic and participatory sensing systems.
- Application of CrowdSenSim for public street lighting, an essential service in current and future smart cities.

The paper is organized as follows. Section II illustrates existing tools for simulation of MCS activities. Section III presents the design criteria of CrowdSenSim highlighting its objectives and scenarios of applicability. Section IV details CrowdSenSim's architecture. Section V presents performance evaluation and Section VI illustrates the use of CrowdSenSim for smart lighting. Finally, Section VII concludes the work and outlines future directions.

II. Background on Crowdsensing

This section reviews works in the field of performance evaluation of MCS systems through simulations. Currently, the existing tools aim either to characterize and model communication aspects or define usage of spatial environment [15]. This section provides a brief summary on this regard in the following paragraphs.

Tanas et al. propose to exploit Network Simulator 3 (NS-3) for crowdsensing simulations [16]. The objective is to assess the performance of a crowdsensing network taking into account the mobility properties of the nodes together with the wireless interface in ad-hoc network mode. Furthermore, the authors present a case study about how participants could report incidents in the public rail transport. NS-3 provides highly accurate estimations of network properties. However, having detailed information on communication properties comes with the cost of losing scalability. First, it is not possible to simulate tens of thousands of users contributing data. Second, the granularity of the duration of NS-3 simulations is typically in the order of minutes. Indeed, the objective is to capture specific behaviors such as the changes of the TCP congestion window. However, the duration of real sensing campaigns is typically in the order of hours or days.

In [17], Farkas and Lendák present a simulation environment developed to investigate performance of crowdsensing applications in an urban parking scenario. Although the application domain is only parking-based, the authors claim that the proposed solution can be applied to other crowdsensing scenarios. However, the scenario considers only drivers as type of users and users travel from one parking spot to another one. The authors consider humans as sensors that trigger parking events. However, to be widely applicable, a crowdsensing simulator has to take into account data generated from mobile and IoT devices' sensors carried by human individuals.

Mehdi et al. propose CupCarbon [18], which is a discrete-event wireless sensor network (WSN) simulator for IoT and smart cities. One of the major strengths is the possibility to model and simulate WSN on realistic urban environments through OpenStreetMap. To set up the simulation, the researchers have to individually deploy on the map the various sensors and the nodes such as mobile users, gas and media sensors and base stations. The approach is not suitable for large scale crowdsensing scenarios with thousands of users.

III. CrowdSenSim: Design Principles

This section presents CrowdSenSim in a nutshell, highlighting the principles of the design, its objectives and the scenarios of applicability. Performing simulations in complex environments such as modern cities requires the simulation platform to be scalable, in other words it does not have to limit the researcher in the choice of important parameters such as the simulation period or the number of users. The scalability requirement is essential and permits CrowdSenSim to measure performance of smart city applications. The following paragraphs illustrate in more details the design criteria.

Scalability: For proper operation, MCS systems require a large number of contributors. Therefore CrowdSenSim is designed to take into account participants in the order of tens of thousands that move a in wide realistic urban environment. Each individual can potentially own several mobile and IoT devices. The time dimension is also important. The duration of a sensing campaign can range from hours to days and CrowdSenSim addresses this challenge efficiently. For instance, let us consider 10 000 users producing data with a duration of only 30 minutes per day. Using commonly available sensors on the market such as an accelerometer working at 50 Hz frequency 12 bits long samples, the total amount of generated data by each user would be 1.35 GB. Considering the prolonged duration of user contribution and additional sensors would considerably augment this figure.

Realistic urban environment: CrowdSenSim relies on realistic urban environments, which makes the simulator flexible and easy to be adopted in any city. Furthermore, it allows to perform analysis that provide meaningful insights to municipalities to understand the feasibility and the potentiality of public services employing MCS techniques. Simulations over a grid or a square area as abstraction levels lower the complexity, but do not allow to take into account important features such as movements in real streets and physical obstacles such as buildings. CrowdSenSim incorporates this feature allowing users to include the layout of cities as input.

User mobility: Human mobility is defined as sequences of spatiotemporal user movements. Understanding human mobility in an urban environments is crucial to design mobility patterns that meet social behaviors and scale to the requirements of modern smart cities [19]. CrowdSenSim includes a number of human mobility patterns designed for pedestrian mobility in urban environments.

Costs of Sensing: The sensing activity impacts on the energy budget of the participants' mobile devices. CrowdSenSim is able to capture the energy directly spent for the sensing tasks as well as the energy spent for communications. IoT and mobile devices are equipped with several communication technologies, including 3G/LTE, WiFi and Bluetooth. Each communication technology drains battery of the devices differently and can have associated costs (e.g., users have a limited monthly plan).

IV. CrowdSenSim: the Architecture

The architecture of CrowdSenSim follows the design specifications illustrated in Section III implementing independent modules to characterize the urban environment, the user mobility, the communication and the crowdsensing inputs, which depends on the application and specific sensing paradigm utilized. Fig. 2 graphically shows the relations between the modules, that are explained in details hereafter.

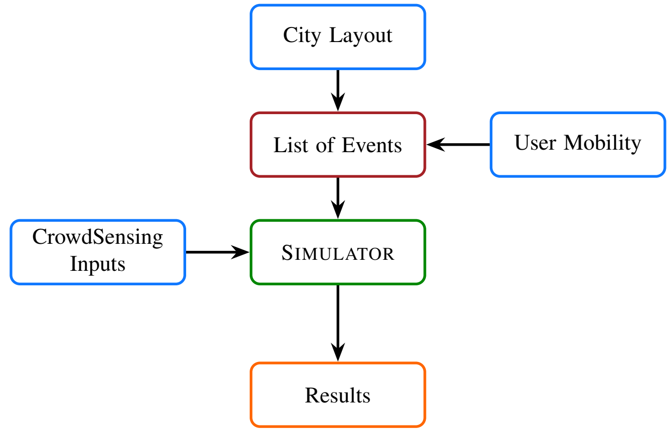


Fig. 2. Main modules of CrowdSenSim

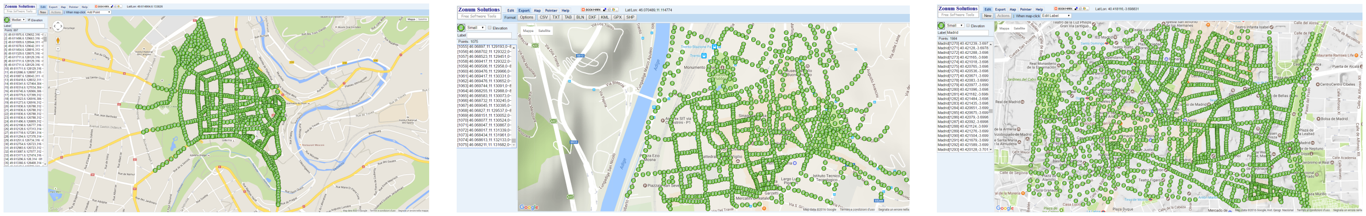
A. City Layout Module

The module in charge of defining the city layout allows the researcher to input into the simulator the city where simulations will be performed. Specifically, the layout of the city is defined in terms of a set of coordinates C containing information on *<latitude, longitude, altitude>*. The set of coordinates compose the streets of the city where the users will move during simulation runtime and can be obtained with online tools like OpenStreetMaps or DigiPoint. In this version of the simulator, we rely on Digipoint, which is a crowdsourced application providing free access to street-level maps [20]. Fig. 3 shows the urban environments currently available for simulations, namely the city center of Luxembourg (see Fig. 3(a)), Trento (see Fig. 3(b)) and Madrid (see Fig. 3(c)). The center of Luxembourg city covers an area of 1.11 km^2 with a population of 110 499 inhabitants as of the end of 2015 and is the home of many national and international institutional buildings. The city center of Trento has a population 117 317 inhabitants as of the beginning of 2016 and is the capital of the homonym Province. The city center of Madrid covers approximately an area of 5.23 km^2 with a resident population of 149 718 residing inhabitants.

The city layout module allows the researcher to define the size of the city and the level of detail of the urban environment. High resolution of the city layout, which corresponds to choose a higher number of coordinates, increases the precision of user movements at the cost of longer and more computationally expensive simulations. Viceversa, a coarse resolution of the city layout makes the simulations to run faster, but lowers the accuracy of users movements and precision of the urban environment. The latter is important: having a high resolution of the urban environment permits to characterize places, e.g., to identify among the others bars, restaurants, schools or hospitals.

B. User Mobility Module

The user mobility module defines the spatiotemporal properties of user movements in the urban environment, which compose the so-called *list of events* (see Fig. 2). We define an *event* as "the arrival of an user in a given coordinate at a given instant of time."



(a) Luxembourg (b) Trento (c) Madrid

Fig. 3. Maps of cities obtained from DigiPoint

The module defines the following steps to determine the spatiotemporal list of events:

- *Inizialization*: it characterizes the *location* and *time* of user arrival.
- *Mobility*: it characterizes the user movements after arrival.

1) Initialization: This initial step is in charge of determining where and when each user starts moving in the city. Each user arrival is therefore characterized by a coordinate c_a and time t_a. In the current version of the simulator, the location is randomly determined among the set of coordinates C of the map. The design choice builds on the assumption that each of the coordinates has the same relevance, i.e., it does not exists a difference between popularity of places. Future implementations will allows the researchers to choose between random and popularity-driven assignment of user location. The time of user arrival can be either randomized or based on real-world traces, which are the results of a study on pedestrian mobility and are public available on Crawdad (ostermalm_dense_run2) [21]. For example, Fig. 4 shows the probability density function of the user arrival resulting from the study of the traces. In practice, to obtain the results presented later in Section V-A2, the density computed in Fig. 4 was scaled to be adapted to an arrival time period between 8:00 AM - 1:40 PM for 20 000 users. The probability density function of user arrival is indeed determined by two global simulation inputs: the total number of users in the system and the simulation period. In random user arrival modes, the default probability density function is uniform, i.e., during the simulation period each minute has the same probability to be chosen as arrival time for each user. The researcher can easily modify the user arrival time by changing the probability density function. In the case study presented in Section VI, we will present a modification of the probability density function of user arrival suitable for the application of public street lighting.

2) Mobility: In the default setting, each user moves over the set of coordinates C for a predefined amount of time T_{move} which is uniformly distributed between $[10, 20]$ minutes with an average speed S_{move} uniformly distributed between $[1, 1.5]$ m/s. The default setting can be easily modified. After arrival in c_a and time t_a, the next move makes the user to *jump* in c_{next} and time t_{next}. The simulator choose c_{next} to be physically in proximity of c_a, i.e., it chooses a coordinate among C which is on the same street or square with distance below a maximum radius. Once determined c_{next}, the simulator computes t_{next} on the basis of the physical distance between c_a and c_{next} the speed of the user. The distance is computed with the Haversine

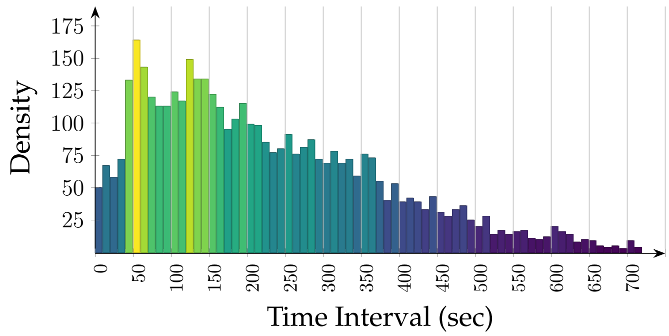


Fig. 4. User distribution of mobility trace "kth/walkers"

formula [22] and permits to compute the amount of time it takes between the two points t_{travel}. Then, t_{next} is determined as follows:

$$t_{\text{next}} = t_a + t_{\text{travel}}, \quad (1)$$

and the total amount of time the user is allows to travel T_{move} is updated as follows:

$$T_{\text{move}} = T_{\text{move}} - t_{\text{travel}}. \quad (2)$$

The user stops moving when $T_{\text{move}} \leq 0$. It is worth to highlight that during each movement the speed of the movement S_{move} changes. The new value is generated again uniformly distributed between $[1, 1.5]$ m/s to mimic the change of velocity during walking.

In the current version, users move only once during the simulation period and it is not possible yet to define a direction of movement for each user. We plan to extend the simulator to take into account this possibility as future work.

C. Crowdsensing Inputs Module

This module defines the inputs specific to crowdsensing analysis. CrowdSenSim relies on two types of inputs. The first set does not depends on the sensing paradigm employed and comprises all the parameters related to sensing and communication operations. The second set includes parameters that are specific to the participatory sensing paradigm. Unlike the opportunistic sensing paradigm which does not have particular input parameters, in participatory systems it is necessary to define the concept of task and how to assign tasks to users.

Sensing and Communication Parameters: In CrowdSenSim, data generation takes into account sensors commonly available in current IoT and mobile devices. Table I presents detailed information on sensors and communication parameters. Specifically, CrowdSenSim generates sensing readings from

TABLE I
SENSOR AND COMMUNICATION EQUIPMENT PARAMETERS USED FOR PERFORMANCE EVALUATION

SENSOR	PARAMETER	VALUE	UNIT
Accelerometer	Sample rate	50	Hz
	Sample size	12	Bits
	Current	35	µA
Temperature	Sample rate	182	Hz
	Sample size	16	Bits
	Current	182	µA
Pressure	Sample rate	157	Hz
	Sample size	16	Bits
	Current	423.9	µA

(a) Sensor Equipment

SYMBOL	VALUE	UNIT	DESCRIPTION
ρ_{id}	3.68	W	Energy in idle mode
ρ_{tx}	0.37	W	Transmission power
ρ_{rx}	0.31	W	Reception power
λ_g	1000	fps	Rate of generation of packets
γ_{xg}	$0.11 \cdot 10^{-3}$	J	Energy cost to elaborate a generated packet

(b) Communication Equipment

the FXOS8700CQ 3axis linear accelerometer from Freescale Semiconductor [23] and the BMP280 from Bosch [24], which is a digital pressure and temperature sensor. For a worst scenario analysis, in the default settings the sensors keep generating data according to their sampling frequency for the entire period of users movements.

For communication purposes, the current version of the simulator employs only WiFi technology. Based on the sample resolution of the sensors, data is first organized in packets of 1 500 Bytes and delivered to the collector continuously during users movements. Each user transmit data to the closes WiFi Access Point (AP). The APs are characterized by *<latitude, longitude>*, not necessarily from the set C. For the city of Luxembourg, the precise location of WiFi APs was obtained from an online tool[1].

Parameters for Participatory Sensing Paradigm: CrowdSenSim defines the following properties for tasks: location, time of deployment, duration and coverage. With the default settings, all the parameters are randomly selected from the set of coordinates C, uniformly distributed within the simulation period and as fraction of the simulation period for location, time of deployment and duration respectively. The task coverage defines the maximum radius where users can actively contribute to the task and is fixed for all the tasks. The researcher can also input to the simulator a file describing the aforementioned properties.

D. Simulator and Results

CrowdSenSim during simulation runtime computes a number of statistics, including energy consumption and amount of data generated and provides the researcher to a visualization tool to display the results. For example, with the help of Google Heatmap tool[2], CrowdSenSim outputs on the real maps the most populated tasks or WiFi APs.

The energy E spent for communication purposes is computed as follows. E is consumed during a transmission time τ_{tx} and is defined as:

$$E = \int_0^{\tau_{tx}} P_{tx} \, \mathrm{d}t, \tag{3}$$

[1]Online: https://www.hotcity.lu/en/laptop/www/About/Wi-Fi-coverage
[2]Available on: https://developers.google.com/maps/documentation/javascript/examples/layer-heatmap

TABLE II
SIMULATION SETTINGS FOR ANALYSIS OF PARTICIPANT RECRUITMENT POLICY

PARAMETER	VALUE
Number of users	[10 000]
Overall evaluation period	8:00 AM - 2:00 PM
Time of travel per user	Uniformly distributed in [10, 20] min
Average user velocity	Uniformly distributed in [1, 1.5] m/s
Timeslot duration	1 minute
Task duration	30 timeslots
Number of tasks	25
$D_{\max}$	30 m

where P_{tx} is the power consumed for transmissions of WiFi packets generated at rate λ_g [25]:

$$P_{tx} = \rho_{id} + \rho_{tx} \cdot \tau_{tx} + \gamma_{xg} \cdot \lambda_g. \tag{4}$$

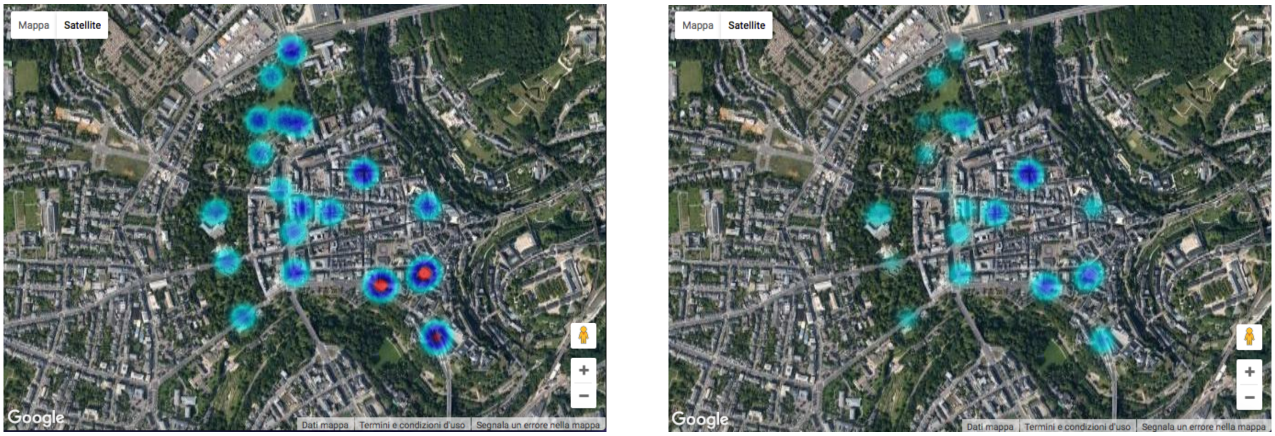
V. PERFORMANCE EVALUATION

This section provides performance analysis of CrowdSenSim. First, the results obtained for participatory and opportunistic sensing systems are illustrated, with a focus on participant recruitment for the former sensing paradigm and energy consumption and amount of data collected for the latter sensing paradigm. Second, technical evaluation of the simulator is shown, with a focus on CPU, processing time and memory utilization.

For performance evaluation, the simulations are carried out using a Unix machine equipped with Ubuntu 14.10. Furthermore, the machine supports a CPU Intel ®Core ™ i3 at 2.27 GHz with a system memory of 1916 MiB.

A. Analysis of Participatory and Opportunistic Crowdsensing Scenarios

1) Participatory Sensing Scenario: In participatory sensing systems, we employ CrowdSenSim in the context of participant recruitment and implement a policy defining user recruitment and task assignment [14]. Devising proper recruitment policy is important. On one hand, it allows the organizer to minimize the expenditure. On the other hand, it helps to choose the users that will carry out the sensing task successfully. For example, in the public safety context, it is essential to select users to maximize the trustworthiness of collected data [26], [27]. The policy can be employed in *distance-based*



(a) SDRP Policy

(b) DBRP Policy

Fig. 5. User recruitment for sensing tasks deployed in Luxembourg

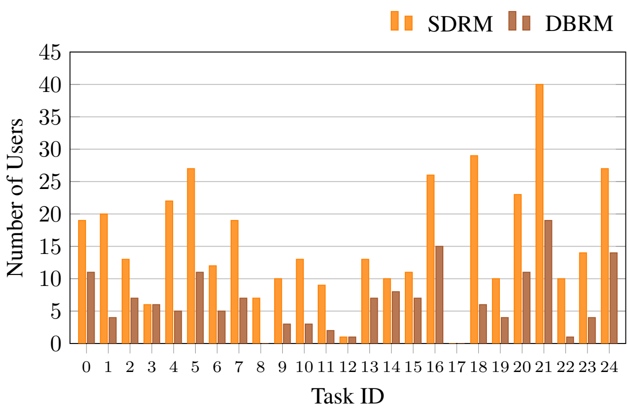


Fig. 6. Number of recruited users under SDRM and DBRM

recruitment mode (DBRM) or *sociability-driven recruitment mode (SDRM)*. In DBRM, the spatial distance between the users and the sensing task is the discriminant factor defining user eligibility. Users far from $D_{\max}$ from the sensing task i are never considered as potential contributors in that task. In SDRM, the user sociability, defined as amount of data users consume or the time they spend using mobile social network applications is the discriminant factor for recruitment.

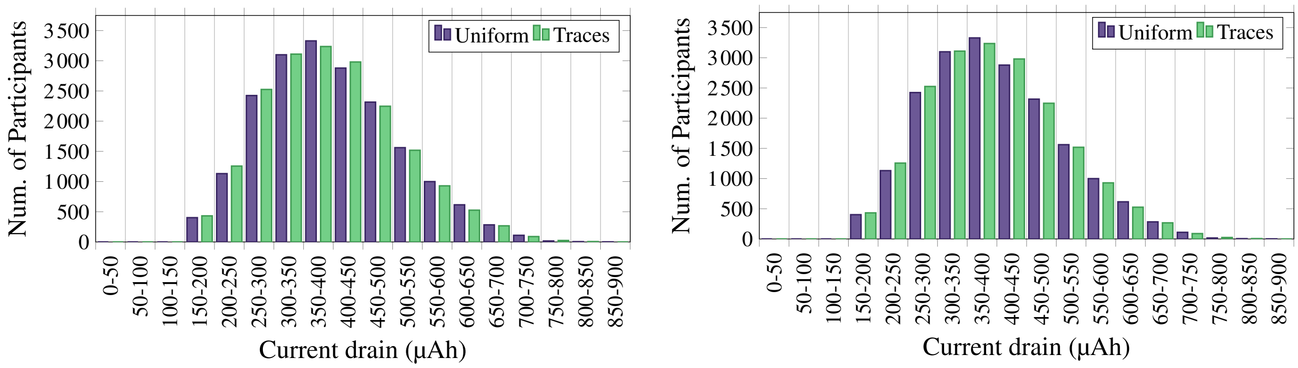
Table II lists the details of the simulation set-up. We employ CrowdSenSim for demonstration purposes to visualize the distribution of user recruitment and refer the reader for further details on the results to [14]. Fig. 5 compares the number of users recruited in SDRM and DBRM for all the deployed 25 tasks in Luxembourg city center using the Google Heatmaps tool. Tasks with higher number of users recruited are marked with a bigger radius and with more bright and intense colors. Fig. 6 shows that SDRM outperforms DBRM as the number of recruited users is higher for all the deployed tasks. Moreover, for task with ID equal to 8, the SDRM is able to recruit users where the DBRM fails.

2) Opportunistic Sensing Scenario: In opportunistic sensing paradigm, users contribute continuously data even if they do not receive a specific task. In this context, CrowdSenSim is employed for evaluation of data generation in the city center of Luxembourg having fixed the number of participants to 20 000, which corresponds to more than one fifth of the population of Luxembourg. The objective of the experiment is to assess during the simulation period from 8:00 AM to 2:00 PM the energy consumption due to sensing and reporting and the amount of generated data under different user arrival pattern. Users move according to the predefined settings illustrated in Section IV-B. In the first user arrival pattern, the start time of the walk is uniformly distributed between 8:00 AM and 1:40 PM to allow users starting moving towards the end of the period to correctly end their journey at 2:00 PM. The second arrival pattern is based on the data set with traces of pedestrian mobility (ostermalm_dense_run2) [21].

Energy Cost for Sensing and Reporting: Fig. 7 presents the distribution of users and their energy spent for sensing with the uniform and traces-based user arrival patterns. As expected, the user arrival pattern does not influence the energy consumption, which only depends on the amount of time the users generate data. As the users contribute data for time periods as low as 10 minutes up to time periods of a maximum of 20 minutes, the profiles of Fig. 7(b) and Fig. 7(a) follows a normal distribution. Current drain of sensing operations is on average 373.41 µAh and 368.80 µAh for uniform and traces-based arrival patterns. In the worst case, few users experience a cost that is nearly more than double with respect to the average. Comparing to the battery capacity available in modern smartphones, which is in the order of 2000 mAh, it is possible to conclude that the energy cost for sensing is negligible with respect to the energy spent for communications (see Fig. 7(b)).

Amount of Data Collected: The amount of information reported by users devices is unveiled in the following experiment, which evaluates the amount of data generated per single sensor for the two different user arrival patterns.

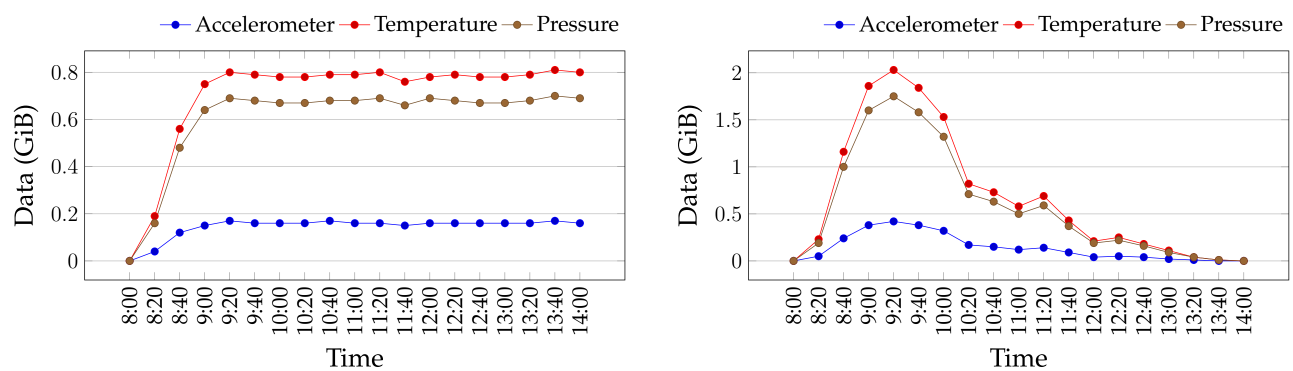
Fig. 8 shows the total amount of data collected along with the simulation period for the two user arrival patterns. As expected, the amount of data is proportional to the sampling frequencies of the three considered sensors. Recalling that each user contributes only during a short period of time (10 to 20 minutes), the amount of collected information is



(a) Sensing Cost

(b) Communication Cost

Fig. 7. Energy spent for sensing and communication



(a) Arrival pattern with uniform distribution

(b) Arrival pattern based on traces

Fig. 8. Amount of data generated

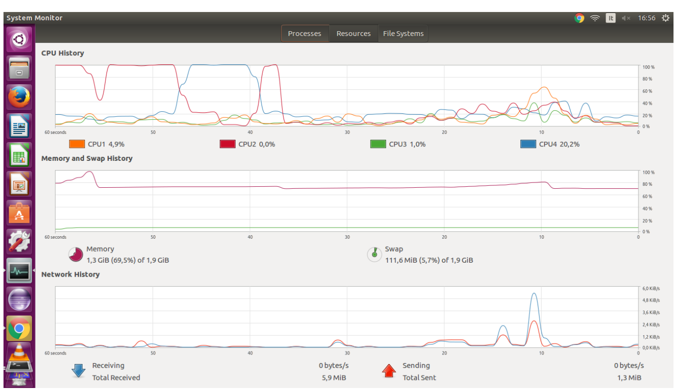


Fig. 9. GNOME System Monitor ***Make it full page otherwise difficult to read***[CF]

considerable. For example, 20 000 users arriving according to the uniform arrival pattern would generate 2.61797 GB, 12.71 GB and 10.96 GB for the accelerometer, temperature and pressure sensors respectively. Fig. 8(a) shows the results for the uniformly distributed arrival pattern. As expected, the amount of contribution remains constant after the initial set up as the amount of users arriving in a given time window is constant along the simulation period. Fig. 8(b) illustrates the results for the user arrival pattern based on the data set. Unlike the previous case, the shape of the curve follows the probability density function of the traces as per Fig. 4.

B. Performance of the Simulator

This section provides a technical evaluation of the simulator performance. The metrics evaluated concern processing time, CPU and memory utilization. The experiments are carried out deploying CrowdSenSim in a Virtual Machine (VM) running Ubuntu 14.10 with two different profile settings, namely 1024 MiB and 2048 MiB of memory. The setting allows us to profile the performance of the simulator perceived by the end users. The VM is equipped with GNOME System Monitor which permits to verify the system performance. Fig. 9 shows an example for a simulation with 20 000 participants in opportunistic sensing scenario.

Fig. 10 shows the profile of the CPU utilization expressed in percentage obtained with the `dstat` tool[3]. The experiment analyzes the performance in a scenario with a huge number of users, 100 000, in the city of Luxembourg. The statistics obtained have been filtered to spot the profile of the process running the simulation. The resulting graph shows that the CPU utilization can occupy as much as 25% of the available resources and this happens at the beginning where most of the computation occurs to process the events.

The next set of experiments aims at assessing the performance of processing time and memory occupancy. Fig. 11 shows the results obtained Both experiments were performed for the city of Luxembourg, with both VMs configurations

[3] Available on: http://dag.wiee.rs/home-made/dstat/

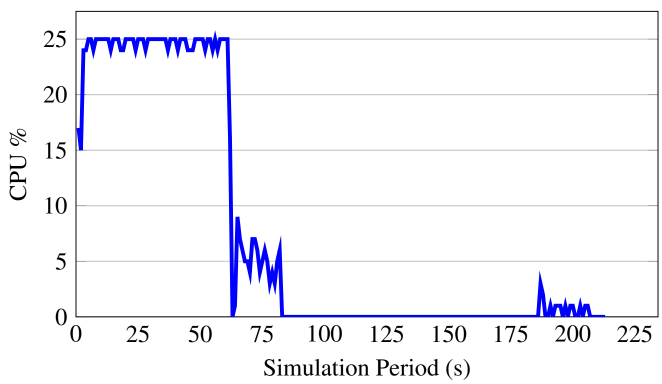


Fig. 10. CPU utilization for a simulation run with 100 000 users

and with an increasing number of participants from the set $\{1\,000, 5\,000, 10\,000, 20\,000, 50\,000, 70\,000, 100\,000\}$. The maximum number of users was selected consistently with the population of the city. Fig. 11(a) analyzes the processing time, which remains almost constant for a number of participants lower than 10 000 and then it increases exponentially for both the configuration settings. Fig. 11(b) analyzes the memory consumption with a focus on the Resident Set Size (RSS), which defines the amount of memory the process occupies in the RAM. For both configurations of the VM, the RSS remains almost identical for a number of participants lower than 20 000, then the process tends to occupy as much as possible all the available resources.

VI. Case study: Smart Lighting

CrowdSenSim is a candidate tool for analysis of smart city services. This section presents a case study where the simulator is employed to assess the performance of public street lighting. However, the capabilities of the simulator are not restrained to this particular application scenario. We are currently working to extend the simulator capabilities to include vehicles as contributors to the data collection process and to analyze other important and challenging issues of modern cities, e.g., waste management. Waste management involves the whole process of monitoring waste locations, truck routes, collection phases and waste disposal. In this context, CrowdSenSim can verify the performance of a monitoring system in which can actively cooperate in locating and rating the situation of sites to identify where collection is urgent.

A. The problem of Smart Lighting in Modern Smart Cities

Public lighting is a traditional city service provided by lampposts widely distributed in streets and roads. Lighting causes nearly 19% of worldwide use of electrical energy and entails a 6% of global emissions of greenhouse gases. A decrease of 40% of energy spent for lighting purposes is equivalent to eliminate half of the emissions from the production of electricity and heat generation of the US [28]. Specifically, public street lightning, which is an essential community service, impacts for around 40% on the cities' energy budget. Consequently, in preparation of the EU commitments, optimizing the lighting service is a primary objective for the municipalities [29].

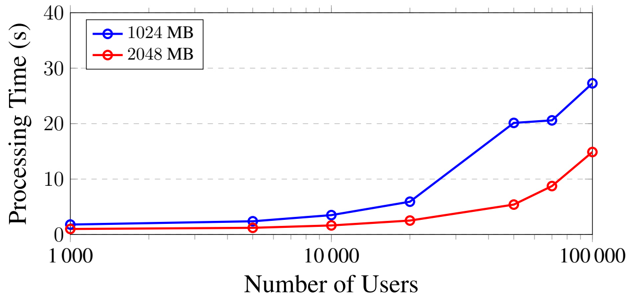
The street lighting solutions currently implemented in cities are not energy efficient. Typically, every lamp operates at full intensity 12 hours a day on average: 8 hours during summer and 14 hours during winter period [29]. As a result, the costs the municipalities sustain are high [28]. A number of different types of lamps are applicable for public street lighting, including High Pressure Sodium (HPS), Metal-halide (MH) lamps, Compact Fluorescent lamps (CFL) and Light-emitting diode (LED). LEDs have an average lifetime 4 times longer than HPS lamps and 10 times longer if compared to MH lamps. Installing LEDs is effective to reduce hardware, installation and maintenance costs. Low wattage provides significant energy savings and allows increasing the lamp efficiency [30], [31]. The HPS lamps do not support dimming and only LEDs can be employed to perform dimming properly. The use of LEDs is gradually gaining popularity due to its photo metric characteristics, such as low weighted energy consumption (kW/1000hrs), high luminous efficacy (lm / W), high mechanical strength, long lifespan and reduction of light pollution. LED lamps can dim the light intensity by more than 50% modifying therefore the output level of light according to the circumstances. For example, when traffic is low or in rarely visited areas of the city, like the parks at night. The city of Brittany in France, dims street lights by 60% between 11 PM and 5 AM to decrease waste energy [29].

We devise a smart lighting method for smart cities which dims the light of lampposts in proportion to the number of users in the vicinity. To detect the presence of users nearby the lampposts a presence sensor like the SE-10 PIR motion sensor is assumed to be installed on site [32]. With presence sensors, every lamppost is able to recognize the presence of citizens within a certain radius R like illustrated in Fig. 12. Similarly to the solution adopted in Brittany, i.e., the minimum light intensity level is 60% if no users are within the coverage radius R and increases or decreases proportionally on the basis of the passage of the users. In more details, if the number of users is increasing, then the light intensity increases or remains at 100%, while if the number of users reduces from previous status, then the light intensity reduces until it reaches the minimum level.

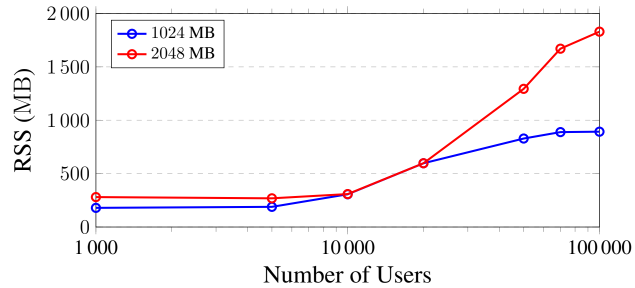
B. Evaluating Smart Lighting Solutions with CrowdSenSim

To evaluate the proposed smart lighting solution with CrowdSenSim, a set of 537 lampposts has been deployed according to their physical location in the streets and squares of Luxembourg City. Fig. 13 details the position of each lamppost given in terms of coordinates *<latitude, longitude, altitude>*. Each lamp is equipped with LED technology and at full light intensity consumes 82.7 kW/1000hrs.

The number of users moving in the city is set to 5 000. Each of them walks for a period of time that is uniformly distributed between $[10, 20]$ minutes with an average speed uniformly distributed between $[1, 1.5]$ m/s. The users begin walking according to a specific arrival pattern. During the evaluation period, set between 9 PM and 7 AM, each user has



(a)



(b)

Fig. 11. Analysis of a) processing time and b) memory with increasing number of users

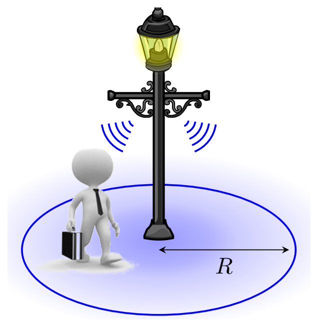


Fig. 12. Coverage radius R

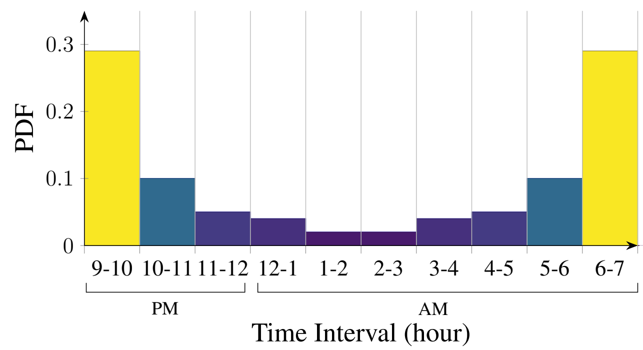


Fig. 14. Probability density function of user mobility during the evaluation period

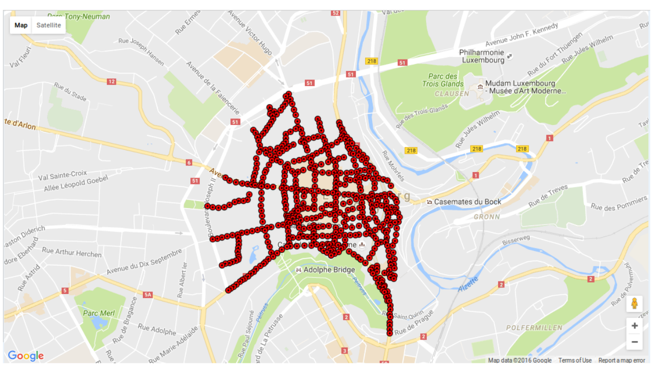


Fig. 13. Position of lampposts in Luxembourg city center

(a) Lampposts activity with LEDs

(b) Lampposts activity with current technology method

Fig. 15. Heatmap of lampposts activity

a probability to start traveling that is defined by the probability density function (PDF) illustrated in Fig. 14. In more details, during 9 PM and 10 PM nearly one third of the total number of users starts walking and at 7 AM all 5 000 users end traveling.

Fig. 15 shows the results of the lamppost activity obtained through CrowdSenSim. On average, the smart lighting solution with LED technology and light dimming saves nearly 68% of energy consumption with respect to the current adopted solution. Indeed, the set of lampposts consumes on average 298.5 and 927.4 kWh per day respectively.

VII. Conclusion

In this paper we presented CrowdSenSim, a simulation platform for MCS systems. CrowdSenSim is tailored to assess sensing activities in large-scale realistic urban environments and is designed to output results on participant recruitment, data generation and the cost sustained for sensing and reporting from the users point of view. We also demonstrated the suitability of the simulator for analysis of smart city services with a case study on public street lighting.

For future work, we plan to implement a more sophisticated and accurate communication model to analyze in details the networking aspects of MCS systems and to exploit CrowdSenSim to investigate other important city services such as smart waste management. Moreover, future efforts will be devoted to extend CrowdSenSim to vehicular environment, where vehicles contribute to the process of data generation in addition to mobile devices. The current trend sees automotive companies to increase on-board equipment of vehicles with storage, computing capabilities and a growing set of sensors. Data collected by these sensors is not only beneficial for the operation of the vehicles and monitoring of their status, but is projected to become a precious source of information for municipalities as well.

References

[1] N. B. Grimm, S. H. Faeth, N. E. Golubiewski, C. L. Redman, J. Wu, X. Bai, and J. M. Briggs, "Global change and the ecology of cities," in *Science*, vol. 319, no. 5864, 2008, pp. 756–760. [Online]. Available: http://www.sciencemag.org/content/319/5864/756.abstract

[2] H. B. Dulal and S. Akbar, "Greenhouse gas emission reduction options for cities: Finding the "coincidence of agendas" between local priorities and climate change mitigation objectives," *Habitat International*, vol. 38, pp. 100 – 105, 2013.

[3] A. Caragliu, C. Del Bo, and P. Nijkamp, "Smart cities in europe," *Journal of urban technology*, vol. 18, no. 2, pp. 65–82, 2011.

[4] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb 2014.

[5] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by Internet of Things," *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 1, pp. 81–93, 2014.

[6] A. D. Cartier, D. H. Lee, B. Kantarci, and L. Foschini, "IoT-Big Data Software Ecosystems for Smart Cities Sensing: Challenges, Open Issues, and Emerging Solutions," in *4th International Workshop on Cloud for IoT (CLIoT)*, 2016, p. (accepted).

[7] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, Fourth quarter 2015.

[8] G. Cardone, A. Cirri, A. Corradi, L. Foschini, R. Ianniello, and R. Montanari, "Crowdsensing in urban areas for city-scale mass gathering management: Geofencing and activity recognition," *IEEE Sensors Journal*, vol. 14, no. 12, pp. 4185–4195, Dec 2014.

[9] J. G. P. Rodrigues, A. Aguiar, and J. Barros, "Sensemycity: Crowdsourcing an urban sensor," *CoRR*, vol. abs/1412.2070, 2014.

[10] A. Antonić, V. Bilas, M. Marjanović, M. Matijašević, D. Oletić, M. Pavelić, I. P. Žarko, K. Pripužić, and L. Skorin-Kapov, "Urban crowd sensing demonstrator: Sense the Zagreb air," in *22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Sept 2014, pp. 423–424.

[11] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, November 2011.

[12] W. Khan, Y. Xiang, M. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 402–427, First 2013.

[13] Google Inc., "Google science journal," 2016. [Online]. Available: https://makingscience.withgoogle.com/science-journal/

[14] C. Fiandrino, B. Kantarci, F. Anjomshoa, D. Kliazovich, P. Bouvry, and J. Matthews, "Sociability-Driven user recruitment in mobile crowdsensing Internet of Things Platforms," in *IEEE Global Communications Conference (GLOBECOM)*, Dec 2016.

[15] K. Farkas and I. Lendák, "Evaluation of simulation engines for crowdsensing activities," in *3rd International Conference & Workshop Mechatronics in Practice and Education*, ser. MECHEDU, 2015, pp. 126–131.

[16] C. Tanas and J. Herrera-Joancomartí, "Crowdsensing simulation using ns-3," *Citizen in Sensor Networks: Second International Workshop, CitiSens 2013*, pp. 47–58, 2014, Springer International Publishing.

[17] K. Farkas and I. Lendák, "Simulation environment for investigating crowd-sensing based urban parking," in *International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, June 2015, pp. 320–327.

[18] K. Mehdi, M. Lounis, A. Bounceur, and T. Kechadi, "Cupcarbon: A multi-agent and discrete event wireless sensor network design and simulation tool," in *7th International ICST Conference on Simulation Tools and Techniques*, ser. SIMUTools '14, 2014, pp. 126–131.

[19] K. K. Jahromi, M. Zignani, S. Gaito, and G. P. Rossi, "Simulating human mobility patterns in urban areas," *Simulation Modelling Practice and Theory*, vol. 62, pp. 137 – 156, 2016.

[20] Zonum SOlutions, "Digipoint 3," 2014. [Online]. Available: http://www.zonums.com/gmaps/digipoint.php

[21] S. T. Kouyoumdjieva, Ólafur Ragnar Helgason, and G. Karlsson, "CRAWDAD dataset kth/walkers (v. 2014-05-05)," Downloaded from http://crawdad.org/kth/walkers/20140505, May 2014.

[22] C. Robusto, "The cosine-haversine formula," *The American Mathematical Monthly*, vol. 64, no. 1, pp. 38–40, 1957.

[23] "FXOS8700CQ: Digital Sensor - 3D Accelerometer + 3D Magnetometer," http://cache.nxp.com/files/sensors/doc/data_sheet/FXOS8700CQ.pdf?pspll=1, 2015.

[24] "BMP280, Barometric Pressure Sensors," https://www.bosch-sensortec.com/bst/products/all_products/bmp280, 2015.

[25] A. Garcia-Saavedra, P. Serrano, A. Banchs, and G. Bianchi, "Energy consumption anatomy of 802.11 devices and its implication on modeling and design," in *8th ACM International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT, 2012, pp. 169–180.

[26] B. Kantarci and H. T. Mouftah, "Trustworthy sensing for public safety in cloud-centric Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 360–368, Aug 2014.

[27] M. Pouryazdan, C. Fiandrino, B. Kantarci, D. Kliazovich, T. Soyata, and P. Bouvry, "Game-theoretic recruitment of sensing service providers for trustworthy cloud-centric Internet-of-Things (IoT) applications," in *IEEE Global Communications Conference (GLOBECOM) Workshops: Fifth International Workshop on Cloud Computing Systems, Networks, and Applications (CCSNA)*, Dec 2016.

[28] M. Castro, A. J. Jara, and A. F. G. Skarmeta, "Smart lighting solutions for smart cities," in *27th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, March 2013, pp. 1374–1379.

[29] "The Business Case for Smart Street Lights," http://www.silverspringnet.com/wp-content/uploads/SilverSpring-Whitepaper-Smart-Street-Light-Bizcase.pdf, 2013, Accessed July 26, 2016.

[30] "Philips ClassicStreet," http://download.p4c.philips.com/lfb/f/fp-912300023301/fp-912300023301_pgl_en_aa_001.pdf, 2016, Accessed July 26, 2016.

[31] "LucaloxTM Standard," http://www.gelighting.com/LightingWeb/ru/images\/HPS_Lucalox_Lamps_Data_sheet_EN.pdf, 2016, Accessed July 26, 2016.

[32] F. Leccese, "Remote-control system of high efficiency and intelligent street lighting using a ZigBee network of devices and sensors," *IEEE Transactions on Power Delivery*, vol. 28, no. 1, pp. 21–28, Jan 2013.

Claudio Fiandrino (S'14) is a postdoctoral researcher at IMDEA Networks Institute, Madrid, Spain. Claudio obtained his Ph.D. degree at the University of Luxembourg working on the ECO-CLOUD project focusing on energy efficient communications in cloud, mobile cloud and fog computing. He received the Bachelor Degree in Ingegneria Telematica in 2010 and the Master Degree in Computer and Communication Networks Engineering in 2012 both from Politecnico di Torino. Claudio's work on indoor localization over fog computing platforms received the Best Paper Award in IEEE CloudNet 2016. Claudio was a Visiting Ph.D. Student for three months at Clarkson University, NY, USA. He served as Publication and Web Chair at the IEEE International Conference on Cloud Networking (CloudNet 2014) and as TPC member in IEEE CloudNet 2014, IEEE ICC 2016 - SAC Cloud Communications and Networking and IEEE EmergiTech 2016. His primary research interests include mobile cloud/fog computing, mobile crowdsensing and data center communication systems.

Andrea Capponi (S'16) is a Ph.D. student at the University of Luxembourg. He received the Bachelor Degree in Telecommunication Engineering and the Master Degree in Telecommunication Engineering both from the University of Pisa, Italy. His primary research interests are Mobile Crowdsensing, Internet of Things and Smart Cities.

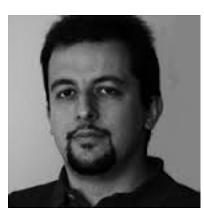
Giuseppe Cacciatore is currently a MS student in Telecommunications Engineering at the University of Trento. He performed his master thesis as an intern at the University of Luxembourg.

Dzmitry Kliazovich (M'03-SM'12) is a Head of Innovation at ExaMotive. He was a Senior Scientist at the Faculty of Science, Technology, and Communication of the University of Luxembourg. Dr. Kliazovich holds an award-winning Ph.D. in Information and Telecommunication Technologies from the University of Trento (Italy). Dr. Kliazovich is a holder of several scientific awards from the IEEE Communications Society and European Research Consortium for Informatics and Mathematics (ERCIM). His works on cloud computing, energy-efficiency, indoor localization, and mobile networks received IEEE/ACM Best Paper Awards. He coordinated organization and chaired a number of highly ranked international conferences and symposia, including the IEEE International Conference on Cloud Networking (CloudNet 2014). Dr. Kliazovich is the author of more than 100 research papers. He is the Associate Editor of the IEEE Communications Surveys and Tutorials and of the IEEE Transactions of Cloud Computing journals. He is a Vice Chair of the IEEE ComSoc Technical Committee on Communications Systems Integration and Modeling. Dr. Kliazovich is a coordinator and principal investigator of the Energy-Efficient Cloud Computing and Communications initiative funded by the National Research Fund of Luxembourg. His main research activities are in the field of intelligent transportation systems, telecommunications, cloud computing, and Internet of Things (IoT).

Ulrich Sorger Ulrich Sorger is a professor at the Computer Science and Communication research unit of the Faculty of Science, Technology and Communication at the University of Luxembourg. His research interest include optimization, coding and information theory. Prof. Sorger holds a Habilitation in Communication Theory and a Ph.D. in Electrical Engineering from the Technical University of Darmstadt.

Pascal Bouvry is a professor in the Computer Science and Communication research unit of the Faculty of Science, Technology and Communication at the University of Luxembourg and a faculty member at the Luxembourg Interdisciplinary Center of Security, Reliability, and Trust. His research interests include cloud & parallel computing, optimization, security and reliability. Prof. Bouvry has a Ph.D. in computer science from the University of Grenoble (INPG), France. He is on the IEEE Cloud Computing and Elsevier Swarm and Evolutionary Computation editorial boards. He is also acting as communication vice-chair of the IEEE STC on Sustainable Computing and co-founder of the IEEE TC on Cybernetics for Cyber-Physical Systems. A full biography is available on page http://pascal.bouvry.org. Contact him at pascal.bouvry@uni.lu.

Burak Kantarci (S'05–M'09–SM'12) is an Assistant Professor with the School of Electrical Engineering and Computer Science at the University of Ottawa (Ontario, Canada). From 2014 to 2016, he was an Assistant Professor at the ECE Department at Clarkson University, where he currently holds a courtesy appointment as an assistant professor. Dr. Kantarci received the M.Sc. and Ph.D. degrees in computer engineering from Istanbul Technical University, in 2005 and 2009, respectively. He received the Siemens Excellence Award in 2005 for his studies in optical burst switching. During his Ph.D. study, he studied as a Visiting Scholar with the University of Ottawa, where he completed the major content of his thesis. He has co-authored over 100 papers in established journals and conferences, and contributed to 11 book chapters. He is the Co-Editor of the book entitled Communication Infrastructures for Cloud Computing. He has served as the Technical Program Co-Chair of seven international conferences/symposia/workshops. He is an Editor of the IEEE Communications Surveys and Tutorials. He also serves as the Secretary of the IEEE ComSoc Communication Systems Integration and Modeling Technical Committee. He is a member of the ACM and a senior member of the IEEE.



Fabrizio Granelli (M'01–SM'05) is currently an Associate Professor and the Delegate for Education with the Department of Information Engineering and Computer Science, University of Trento, Italy. He was the IEEE ComSoc Distingushed Lecturer from 2012 to 2015. He is currently the ComSoc Director for Online Content. He has authored over 170 papers published in international journals, books and conferences in networking, with specific emphasis on wireless networks, cross-layer optimization, cognitive radios and networks, and software- defined networking.

Stefano Giordano (SM'10) Stefano Giordano received the master's degree in electronics engineering and the Ph.D. degree in information engineering from the Università di Pisa, Pisa, Italy, in 1990 and 1994, respectively. He is currently an Associate Professor with the Department of Information Engineering, Università di Pisa, where he is responsible for the telecommunication networks laboratories. His research interests are telecommunication networks analysis and design, simulation of communication networks, and multimedia communications. He is a member of the Editorial Board of the IEEE COMMUNICATION SURVEYS AND TUTORIALS. He was the Chair of the Communication Systems Integration and Modeling Technical Committee. He is an Associate Editor of the International Journal on Communication Systems and the Journal of Communication Software and Systems technically cosponsored by the IEEE Communication Society. He was one of the referees of the European Union, the National Science Foundation, and the Italian Ministry of Economic Development.