Energy Efficient and Quality-Driven Continuous Sensor Management for Mobile IoT Applications

Lea Skorin-Kapov, Krešimir Pripužić, Martina Marjanović, Aleksandar Antonić and Ivana Podnar Žarko
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3, Zagreb, Croatia
Email(s): {lea.skorin-kapov, kresimir.pripuzic, martina.marjanovic, aleksandar.antonic, ivana.podnar}@fer.hr

***Abstract*—A novel class of mobile Internet of Things applications falls under the category of mobile crowdsensing, whereby large amounts of sensed data are collected and shared by mobile sensing and computing devices for the purposes of observing phenomena of common interest (e.g., traffic monitoring, environmental monitoring). Challenges arise with respect to collecting and managing sensor data in an energy- and bandwidth-efficient manner. In this paper we present a cloud-based system architecture centred around a publish/subscribe middleware interfaced with a quality-driven sensor management function, applicable for building mobile IoT applications. The architecture is designed so as to smartly manage and acquire sensor readings in order to satisfy global sensing coverage requirements, while obviating redundant sensor activity and consequently reducing overall system energy consumption. We evaluate the system using a proposed model for calculating bandwidth and energy savings. Model evaluation based on simulation results provides insight into the energy savings for different application requirements and geographical sensor distribution scenarios. Our results show that in certain identified cases, significant energy consumption reductions can be achieved utilizing the proposed architecture and sensor management scheme (as compared to a standard publish/subscribe approach), while maintaining overall global sensing quality level (in terms of required sensing coverage). Assumptions with regards to user distributions in urban areas are verified using an existing dataset reported in literature.**

I. Introduction

With the proliferation of mobile sensing and computing devices (e.g., smartphones) equipped with various embedded sensors and network communication capabilities, such devices are becoming a potential source of large amounts of sensing data at the edge of the Internet. A wide range of new applications utilizing such data have been envisioned across a number of domains, such as healthcare, transportation, environmental monitoring, smart homes, and social networks [1]. Billions of objects are expected to take on the role of providing data from the physical world to the Internet [2]. While certain application scenarios focus around data collected by/for a single individual for personal reasons (e.g., personal health monitoring), an emerging category of applications are based on the collection of sensing data at a community-wide level, whereby multiple individuals provide sensing data in order to contribute to the observation of a large scale phenomena, such as for example the occurrence of traffic congestion, or environmental pollution and temperature monitoring.

The notion of *Mobile Crowdsensing* (MCS) has been introduced to refer to a category of applications where individuals utilizing mobile devices "collectively share data and extract information to measure and map phenomena of common interest" [3]. Such applications have been further categorized as either participatory [4] (assuming the active involvement of participants in choosing to contribute data), and opportunistic [1] (referring to autonomous data collection, not requiring explicit user interaction).

While a number of challenges arise in the context of deploying MCS applications, we focus this paper around three main issues. First of all, battery-operated mobile devices and sensors suffer from a limited battery lifetime. Hence, there is a need for solutions that will limit the energy consumptions of such mobile Internet-connected objects (ICOs) while at the same time providing sufficient sensing coverage. Secondly, collecting and transmitting large amounts of sensed data over wireless network interfaces imposes a heavy burden in terms of network bandwidth consumption. Consequently, solutions are needed that will optimize the amount of data transmitted over the network in accordance with the needs of users/applications in terms of sensor data readings. For example, if certain sensor readings are not of interest, or redundant, filtering mechanisms may be applied at the network edges. In a mobile crowdsensing scenario, the mobility of users and ICOs leads to the dynamic sensing coverage of geographical areas, potentially leading to certain areas being sufficiently/redundantly covered, while other areas may suffer from lack of available data. As a third issue, efficient data processing mechanisms are needed to handle large amounts of collected data, closely linked to the problem of Big Data processing in the cloud [5].

In our previous work [6], [7], we have proposed a CloUd-based PUblish/Subscribe (CUPUS) middleware solution for MCS applications, supporting selective real-time acquisition and filtering of sensor data on mobile devices (hence addressing energy- and bandwidth-efficiency), efficient continuous data processing in the cloud, and near real-time delivery of sensor data to mobile devices. Data is collected from mobile devices over widespread geographical areas using an *opportunistic sensing* approach, i.e., sensor data is acquired autonomously and reported to a cloud platform without explicit user intervention, typically periodically. The solution has been developed within the scope of the EU FP7 OpenIoT project (Open Source blueprint for large scale self-organizing cloud environments)[1], and demonstrated via an MCS air quality monitoring application.

[1] http://openiot.eu/

978-1-63190-043-3 © 2014 ICST

DOI 10.4108/icst.collaboratecom.2014.257320

While energy-efficient and flexible data acquisition mechanisms have been previously addressed, in this paper we build on this previous work by introducing support for **quality-driven** continuous sensor management, i.e. in cases of redundant available sensor data, intelligent decisions are made regarding an optimal subset of available sensors which to keep active in order to meet subscription requirements. Such decisions are made by a *QoS sensor management component* and based on parameters such as sensor accuracy, level of trustworthiness, and available battery level. The overall system architecture and communication infrastructure are evaluated based on simulation results using a proposed model for calculating energy savings. The model assumptions which are made with regards to user distributions in urban areas are verified using an existing dataset reported in literature. Obtained results provide insight into the energy savings for different application requirements (in terms of sensor coverage) and geographical sensor distribution scenarios. Our results show that in certain cases, energy consumption reductions between 40-80% can be achieved utilizing the proposed sensor management scheme, while maintaining overall global sensing quality level (in terms of required sensing coverage).

The rest of the paper is organized as follows: Section II gives a brief overview of related work addressing energy- and bandwidth-efficient data collection. In Section III we present our overall MCS architecture including a cloud-based publish/subscribe middleware and sensor management function. We further discuss a decision-making mechanism for managing sensor data acquisition based on a top k/w approach. Section IV presents an analytical model for calculating energy savings, while Section V presents the results of simulations aimed to test the energy savings based on different input scenarios. Section VI provides concluding remarks and directions for ongoing and future work.

II. Related work

With regards to sensing data acquisition and processing, two general scenarios may be identified. The first deals with situations in which there is a lack of data needed to meet application requirements (e.g., for an environmental monitoring application, there are no available sensor readings in a given geographic area, or for a given time frame). In such cases, techniques based on data interpolation and estimation may be employed. In a second scenario, there may be redundant data available, hence providing opportunities to optimize data acquisition and achieve energy efficiency.

An important issue to address in the deployment of a successful sensor network is the tradeoff between obtaining a sufficient amount of sensor measurements in order to meet existing application requirements, and achieving energy efficiency in order to extend the overall lifetime of the network [8]. The authors in [9] provide an extensive overview of utility-driven data acquisition techniques for efficient collection of data in participatory sensing, whereby queries of different types (e.g., one-shot queries, continuous monitoring queries) may come from different applications. Multi-query optimization problems are formulated and heuristics are proposed for providing effective solutions such as maximizing the overall system utility (i.e., social welfare). Utility functions are introduced specifying the difference between the value of given query results, and the cost for obtaining results. In the context of data acquisition, proposed algorithms aim to achieve efficient sharing of sensor data among multiple queries that may be of different types.

Specifically focusing on mobility aspects, MCS applications take the advantage of a population of individuals to measure large-scale phenomenon that cannot be otherwise measured by individuals [10]. The challenges of meeting resource limitations in the context of MCS applications are summarized in [3]. The authors further discuss resource allocation challenges in the case of multiple concurrent applications sampling various sensors on a single mobile device. Potential solutions include prioritizing applications that require sensor data, hence reducing or increasing the sampling rate of certain sensors while aiming to achieve efficient energy consumption of the mobile device.

A discussion of different MCS applications and optimizing smartphone related energy consumption is given in [11]. Within the scope of the NSF-funded project Citisense, a participatory air quality sensing system has been developed that collects data from body-worn sensor boards which relay the sensor readings to the wearers mobile phone, both for display to the user and relay to a back-end server for a variety of uses, such as the inference of a more detailed regional air quality map [12], [13]. Such an application provides the opportunity to more accurately model air quality that captures microenvironment variations, as compared to data collected only from stationary government monitoring sites. In addition to Citisense, another project addressing environmental exposure feedback systems in the form of mobile participatory sensing is the Common Sense project [14].

In [12], the authors address the problem of energy efficiency and present an approach for model-driven adaptive environmental sensing. They focus on reducing the amount of redundant sensor readings, consequently reducing the amount of communication between client devices and a back-end data collection server. Mobile devices maintain local models of expected sensor readings, hence generate predictive readings, and push updates to the back-end server only in cases when predicted values do not match actual sensor readings. We note that such an approach could potentially be considered complementary to the energy savings mechanisms proposed in the scope of this paper. Similar work also addressing the use of predictive probabilistic models to minimize energy consumption in wireless sensor networks has been previously reported in a highly cited paper by [15].

Further in the scope of the Citisense project, studies reported on efficient energy management and data recovery [8], where authors describe how they leverage correlations between different types of data sources to dramatically reduce the amount of data sent while still being able to reconstruct all of the data with small and controllable error. Experimental results with simulated wireless channel conditions and data collected from two real-world sensor networks (environmental monitoring application and health monitoring application) show that by sampling only 20% of the data, the remaining 80% can be reconstructed with 9% mean error, and energy savings up to 76%. Both tested sensor networks rely on fixed sensors; hence do not take into account any mobility aspects. Related work has been also reported in [16], leveraging contextual information (both user requests and harvesting energy availability) to intelligently adapt sensor sampling rate.

In the work reported in [10], the authors propose a collaborative mobile sensing framework called Mobile Sensor Data EngiNe (MOSDEN), designed to operate on smartphones and capture and share sensed data between multiple distributed applications and users. The engine is designed so as to be compatible with a GSN (Global Sensor Network) middleware. By supporting processing and storage on end user smartphone devices, the platform aims to reduce the necessary data transmission to a centralized server, consequently achieving bandwidth and energy efficiency. In their subsequent work [17], the authors specifically address sensor discovery and configuration challenges. In that context, they address issues such as configuring sensor sampling rate to determine an optimal balance between user (application) requirements and energy consumption; and determining the frequency of network data transmission, e.g., to a cloud-based IoT platform.

While a number of aforementioned projects and approaches focus on mobile/fixed sensing architectures and address the issues of energy- and bandwidth-efficient data collection, what is missing is a generalized solution for providing quality-driven support for achieving energy efficiency, applicable in particular for cloud-based mobile IoT application scenarios.

III. System architecture

A. Cloud-based publish/subscribe middleware

As stated in the introduction, our previous work has introduced the CUPUS[2] middleware [6], [7], supporting context-aware and energy-efficient acquisition and filtering of sensor data in mobile environments. The CUPUS communication infrastructure is based on a continuous processing and communication model, whereby data sources (referred to as *publishers*) disseminate data using a push-based mechanism to interested data destinations (*subscribers*). Users/applications generate data queries referred to as *subscriptions* [18].

A view of the CUPUS architecture is given in Fig. 1. The central component is the Cloud-based Publish/Subscribe Processing Engine (CPSP Engine), responsible for acquiring data from external data sources (e.g., smartphones) processing the data to see if it matches any active subscriptions, and disseminating the data to external data consumers. The engine performs efficient data processing of continuous sensor readings and data queries, and has the ability to elastically adapt to an incoming publication rate and to scale with an increasing number of components. It is implemented as a multi process Java application running on a server machine. Since the matching of publications to subscriptions is the most demanding task performed by the CUPUS middleware, we replicate the matcher processes and dynamically allocate cloud resources for matching in accordance with the processing load.

The described communication model provides the means for selective acquisition of sensor data from mobile wearable sensors as well as filtering of sensor data on mobile devices prior to its delivery into the cloud for further processing. This is done by way of deploying a component called a *mobile broker* on an external data source. The mobile broker *announces* to the CPSP engine the type of data that may be provided by one or more data publishers (sensors representing a data source) connected to the device running the mobile broker (e.g., via bluetooth). The mobile broker is implemented as an Android application which consists of two background services (i.e. of the sensor and mobile broker services), a GUI for controlling the services and presenting the live data, and the mobile broker component. The main task of the sensor service is to acquire sensor readings from connected sensors. It serves as a publisher which is connected to the mobile broker service. In practice, the communication between the sensor and mobile broker service is implemented through the Android internal intent broadcasting and filtering mechanism.

In contrast to existing centralized database solutions that typically send all sensed data into the cloud, the CUPUS publish/subscribe-based solution enables flexible and controlled data acquisition and its subsequent transmission into the cloud only in situations when the sensed data is indeed required by an application. The filtering process itself is performed on the mobile device and guided from the cloud based on global sensor data requirements.

The system is designed so as to keep track of all available sensors and their current locations. Based on received *Announce* messages, the CPSP Engine knows the locations and characteristics of all available data sources for an MCS application, and can turn them on when their data is needed by sending subscriptions matching defined data types to mobile brokers. A main novelty of CUPUS as compared to existing publish/subscribe solutions is in the implementation of such mobile brokers running on mobile devices. We note that further details regarding the CUPUS communication infrastructure can be found in [7].

B. Quality-driven sensor management support

While the CUPUS architecture supports controlled data acquisition based on global data requirements, the CPSP engine does not provide further intelligent decision-making mechanisms aimed at optimizing sensing data quality and energy consumption, since its main task is to perform efficient matching of sensor data to subscriptions (i.e., continuous queries) and to deliver notifications reporting sensor measurements to distributed destinations (e.g. mobile devices) in near real-time. Cases when redundant sensed data is available may be envisioned (e.g., due to a large number of sensors generating data in a certain geographic area), whereby a subset of sensors may be requested to transmit data, while others may be deactivated. Decisions on determining an optimal subset of sensors which to keep active in order to meet subscription requirements may be made based on parameters such as sensor accuracy, level of sensor trustworthiness, and available battery level.

In order to address the challenges of quality-driven sensor management, we introduce a QoS Sensor Management Function (QSMF) interfacing with the CPSP engine. The QSMF is designed to be responsible for two key functionalities: (1) subscription monitoring and management, whereby the QSMF monitors subscriptions received by the CPSP engine and aggregates multiple sensor data subscriptions to determine

[2]The CUPUS and QoS Sensor Management Function source codes are available at the OpenIoT project's Github - https://github.com/OpenIotOrg/openiot/

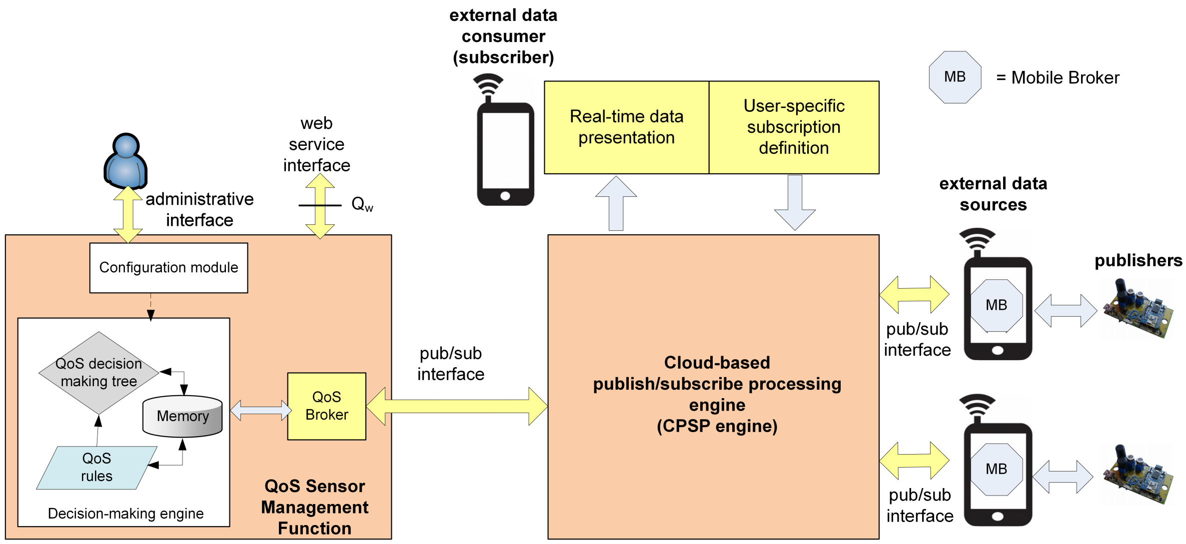

Fig. 1. The CUPUS architecture enhanced with a QoS sensor management function

global application requirements with respect to sensor data acquisition; and (2) data acquisition management, whereby the QSMF manages the sensor data acquisition in order to optimise energy and bandwidth consumption while meeting application requirements.

The QSMF is composed of the following elements:

- Quality of Service Broker (QoS broker): collects information about all available data sources in the CUPUS middleware and all currently active user subscriptions, publishes processed data readings and sends control messages to the CUPUS middleware based on decisions made by a decision making engine.
- Decision making engine: collects and stores target sensor data (specified by current geographic area and mobile broker ID) regarding sensor data accuracy, reliability, and battery level (collected as sensor metadata in *publish* messages) for the purpose of invoking intelligent data acquisition mechanisms. Based on results of the decision process, data acquisition management activities will be invoked, such as activation/deactivation of certain sensors based on application subscription requirements, deactivation of a chosen sensor due to low battery level, or decreased frequency of sensor readings in order to reduce energy consumption.
- Configuration module: offers an administrative interface for configuring the decision making engine (in terms of specifying rules and thresholds, such as minimum number of required sensor readings in a given geographic area).

An example of the quality-driven data acquisition management procedure is shown in Fig. 2. The QSMF is shown as subscribing to a set of mobile sensors that have announced their presence in a given geographic area (hence the QSMF is aware of both their location and collected data). Data is collected and stored by the QSMF and used as input for the decision making process. A number of different conditions may be evaluated

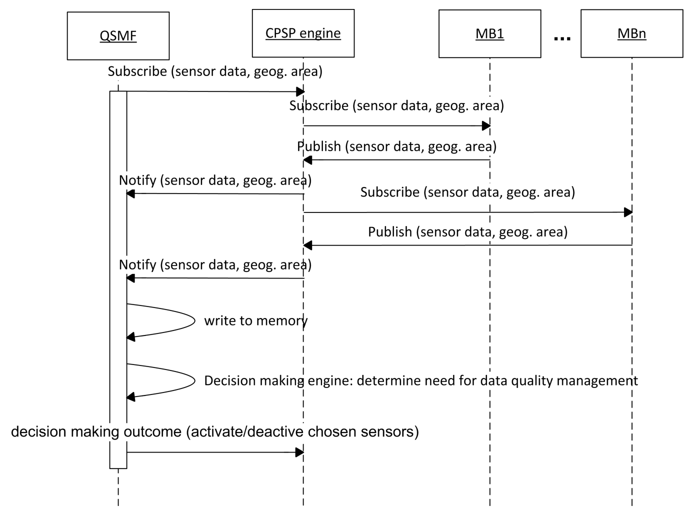

Fig. 2. Quality-driven sensor data acquisition management

(based on the sensor data readings and comparisons) in order to determine the need for data acquisition management, e.g. (i) are there redundant sensor readings available in a given location? (ii) are there insufficient sensor data readings to provide the target level of data accuracy? (iii) are there sensors providing data with large variance as compared to the mean sensor data readings for a given area (hence indicating corrupt data or fault issues)? and (iv) based on sensor data readings, are there sensors providing data falling outside of expected ranges?

Based on evaluated conditions, different decisions may be drawn. For example, given redundant sensor readings, a decision needs to be made as to how many and which sensors may be deactivated, in order to determine a subset of sensors that will meet the current requirements of active applications (determined based on subscriptions) in terms of the number/frequency of necessary sensor readings. These requirements correspond to how many sensor data readings are needed in a given geographic area and over a given time period to adequately respond to the existing application subscriptions, which will depend on the phenomenon being observed and the quality and trustworthiness of the sensors. As an example, consider an air quality monitoring application offering end users the ability to subscribe to the following data: current data regarding air temperature, CO and SO2 for a defined geographic area x; or data regarding average air temperature, CO and SO2 for a defined geographic area x in the time period $[t_1, t_2]$.

In the case of redundant sensors available in a certain area, the basis for making decisions with regards to deactivation of certain sensors is in determining which of the available sensors are of higher value (i.e., compared to other sensors) in terms of their ability to respond to the defined subscriptions with the highest quality. The quality may be directly linked to the the accuracy and sensitivity of the measurements provided by a sensor, as well as the level of trustworthiness (in terms of the trust one can place on a sensor that it will deliver true measurements in time within the scope of its technical parameters). In addition, priority may be given to sensors with a higher remaining battery lifetime, and also sensors that are already active (as opposed to sensors that are available but currently idle) in order to minimize the need for message exchanges (between the CPSP engine, mobile broker, and publisher). Hence, the *values* of different sensors may be calculated using so-called valuation functions (calculated in order to rank sensors from "best" to "worst"), expressed in terms of parameters such as battery level, trust level, and current state of the sensor (active/inactive). Optimization techniques such as those extensively discussed in [9] propose formulating and solving an optimization solution to calculate an optimal sensor allocation scheme in a mobile context taking into account sensor valuation functions. A myopic approach is discussed, in which utility is maximized at a current time slot without considering the future state of the system. Given the highly dynamic nature of mobile sensors, and potential scalability and performance issues related to solving the posed optimization problem, a simplified approach may be adopted.

One possible option is adopting a top-k/w algorithm approach, based on finding the top-k elements (sensors) in a fixed *geographic-window*, i.e., the window corresponds to a specific geographic area. The value of k corresponds to the minimum number of sensors that is needed in a given time interval in order to obtain the required sensor readings. This will clearly depend on the frequency of sensor readings that a given sensor is able to publish. Hence, a decision making process determining the top-k sensors can be used to determine which sensors to activate/deactivate. Consequently, sensors can be ranked based on some form of specified valuation function, as previously discussed. Such a solution assumes a sorted set of sensors according to calculated rank (Fig. 3). Upon a new sensor entering a given geographic area, it is necessary to determine whether it is among the top-k sensors. If so, a sensor at the end of the top-k queue is dropped, i.e. it is deactivated. Analogously, if an active sensor (i.e. belongs to top-k) leaves a designated geographic area, a new sensor will be required to enter the top-k queue.

We note that the focus of this paper is not on exploring the formulation of different sensor valuation functions (certain

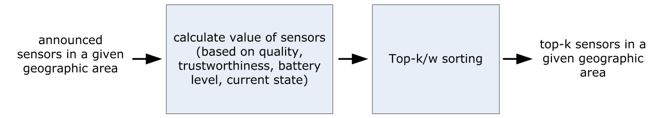

Fig. 3. Top-k/w sensor ranking approach

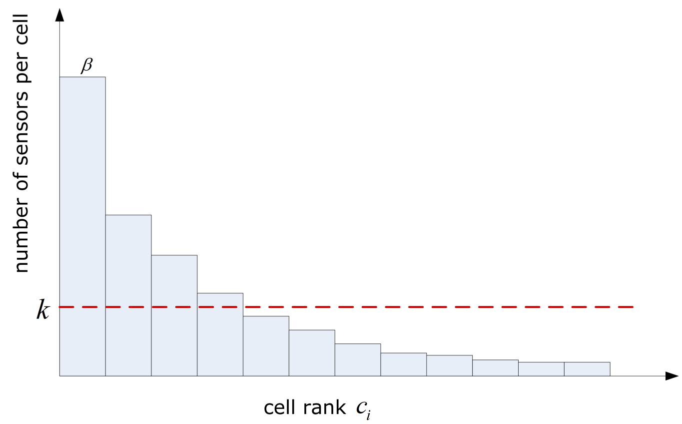

Fig. 4. Example mobile sensor distribution across different cells. Value k refers to the number of required unique sensors publishing data needed to meet the sensing coverage requirements of a given application.

examples can be found in [9]) and evaluating the performance of the aforementioned top-k/w approach, but rather on modeling the energy and bandwidth savings of the CUPUS communication approach (assuming the role of the QSMF) regardless of the methodology used to determine which k sensors (of those announced) to keep active. We therefore focus on modeling and evaluating the energy savings for different application requirements and sensor distributions, and leave further modeling and evaluation of the QSMF decision making engine for future work.

IV. Modeling energy savings of the CUPUS sensor management system

In this section we first present a mobility model for mobile sensors in urban crowdsensing applications. After that we use this model to estimate the number of exchanged messages in a crowdsensing application with and without the CUPUS middleware. This knowledge is then further used to model the potential energy savings achieved by deploying the CUPUS middleware together with the QSMF.

It has been previously shown that the distribution of smartphone users in urban areas (frequency of places visited) follows a power law distribution [19] (confirming also previous findings [20]). Consequently, a large number of areas (referred to as cells) will exist in the 'long tail' of the distribution, i.e., they will be infrequently visited. Therefore, we draw on the assumption that mobile sensors (carried by individuals) are distributed in cells according to a power law distribution such that there are just a few cells with a large number of sensors, while only a small number of sensors are located in the remaining cells. In Fig. 4 we can see such a sensor distribution across different cells, where cells are ranked from left to right according to the number of sensors in each cell. The number k indicates the number of sensors needed to obtain the required sensor readings (for simplification purposes, we assume that sensors generate data at the same frequency). The number of sensors in a cell with rank $c_i = 1, 2, \ldots, C$ is given by the following equation:

$$n(c_i) = \beta \cdot {c_i}^{-\alpha}, \tag{1}$$

where β is the number of sensors in the most populated cell and α is a constant close to 1 which determines the rate of the distribution tail decay.

If N is the total number of sensors across all C cells, parameter β can be calculated as follows:

$$\beta = \frac{n(c_i)}{{c_i}^{-\alpha}} = \frac{\sum_{c_i=1}^{C} n(c_i)}{\sum_{c_i=1}^{C} {c_i}^{-\alpha}} = \frac{N}{\sum_{i=1}^{C} i^{-\alpha}} \tag{2}$$

In our mobility model, the behavior of mobile sensors in a period of time can be described as a finite sequence of S steps: $s_1, s_2, \cdots s_S$. As we assume that sensors are mobile, i.e., some of them will go to a neighboring cell between each two consecutive steps. To quantify the number of moving sensors, we use parameter p_{cng} which defines the probability the a sensor will move to a neighboring cell between any two consecutive steps. Additionally, we assume that sensors which are active always produce exactly m_p publications of sensor readings between each of the consecutive steps.

One of the main characteristics of the CUPUS middleware is that it suppresses redundant and non-relevant sensor readings (i.e. no subscription exists for that reading) by deactivating some of the sensors. In a standard publish/subscribe system, all collected sensor data would be published. Hence, without the CUPUS middleware, all of the N sensors would be active and publish m_p publications in each step. Therefore, in this case the following equation gives the total number of exchanged messages between each two consecutive steps:

$$M_{alternative} = N \cdot m_p. \tag{3}$$

The number of exchanged messages when the CUPUS middleware is used is equal to the sum of *publish*, *announce* and *control* (i.e. activation and deactivation) messages:

$$M_{CUPUS} = M_{publish} + M_{announce} + M_{control}. \tag{4}$$

To estimate the number of exchanged messages when the CUPUS middleware is used, we use the following set of assumptions:

1) the probability that a cell has at least one active subscription between any two consecutive steps is defined as p_{cov},
2) if a cell is covered by at least one subscription between any two consecutive steps then every other more populated cell is also covered in this period (as subscriptions and publications have similar distributions in practice [21]) and
3) the overall sensor distribution across different cells does not change between two consecutive steps.

According to the second assumption, only $p_{cov} \cdot C$ most populated cells are covered with subscriptions between any two consecutive steps. The third assumption means that the distribution of sensors (as depicted in the example in Fig. 4) will remain unchanged between the steps, even if their ranks (according to the popularity) are changed in this period due to sensor mobility. We base this assumption on the fact that we are considering small time steps and cell size. Future empirical data is needed to confirm such assumptions for concrete time intervals and sensor mobility behavior.

In a covered cell with rank c_i, there are $n(c_i)$ sensors which are potential publishers. Since k sensor publishers per cell is adequate for the application scenario, the QSMF will deactivate $c_i - k$ sensors in a cell for which $n(c_i) > k$ and none otherwise. In Fig. 4 we can easily identify cells in which sensors will be deactivated as those that are crossed by the red line which shows parameter k. Therefore, when the CUPUS middleware is used, the number of published messages between each two consecutive steps is equal to the following:

$$M_{publish} = m_p \cdot \sum_{i=1}^{p_{cov} \cdot C} min(\beta \cdot i^{-\alpha}, k). \tag{5}$$

Note that in the previous equation we use the second assumption from the list above such that we assume that only $p_{cov} \cdot C$ most populated cells are covered.

When a sensor moves to a neighboring cell it will send an *announce* message and receive a reply. Therefore, we obtain the number of exchanged messages between each two consecutive steps by multiplying the number of sensors which changed their cells (probability of cell change denoted as p_{cng}) with factor 2:

$$M_{announce} = N \cdot p_{cng} \cdot 2. \tag{6}$$

To calculate the number of control messages we need to know the probability that a sensor is being activated, which is given by the following equation:

$$p_{activated} = \frac{1}{N} \cdot \sum_{i=1}^{p_{cov} \cdot C} min(\beta \cdot i^{-\alpha}, k). \tag{7}$$

When a sensor moves to a new cell in which $n(c_i) > k$, it can become a (new) top-k sensor in that cell and thus a previously active sensor needs to be deactivated. Similarly, if the same condition holds for the source cell, then a deactivated sensor has to be activated. The expected number of control messages that are generated due to a moving sensor is as follows:

$$\begin{aligned} m_c \quad &= \quad 2 \cdot (p_{activated})^2 + 2 \cdot p_{activated} \cdot (1 - p_{activated}) = \\ &\quad\ \ 2 \cdot p_{activated}, \end{aligned} \tag{8}$$

since 2 messages are generated when the sensor is active both in the source and destination cell, 1 message is generated when the sensor is active only in the source cell, 1 message is generated when the sensor is active only in the destination cell and 0 messages are generated when sensor is neither active in the source nor in the destination cell.

Therefore we obtain the number of control messages between each two consecutive steps by multiplying m_c with the number of moving sensors:

$$\begin{aligned} M_{control} &= N \cdot p_{cng} \cdot 2 \cdot p_{activated} = \\ & 2 \cdot p_{cng} \cdot \sum_{i=1}^{p_{cov} \cdot C} min(\beta \cdot i^{-\alpha}, k). \end{aligned} \quad (9)$$

We obtain the number of generated messages between each two consecutive steps when the CUPUS middleware is used from equations (4), (5), (6) and (9) as follows:

$$\begin{gathered} M_{CUPUS} = M_{publish} + M_{announce} + M_{control} = \\ m_p \cdot \sum_{i=1}^{p_{cov} \cdot C} min(\beta \cdot i^{-\alpha}, k) + N \cdot p_{cng} \cdot 2 + \\ 2 \cdot p_{cng} \cdot \sum_{i=1}^{p_{cov} \cdot C} min(\beta \cdot i^{-\alpha}, k) = \\ (m_p + 2 \cdot p_{cng}) \cdot \sum_{i=1}^{p_{cov} \cdot C} min(\beta \cdot i^{-\alpha}, k) + \\ N \cdot p_{cng} \cdot 2. \end{gathered} \quad (10)$$

Finally, we observe the possible energy savings that can be achieved with our approach as compared to a baseline publish/subscribe approach. Given that the energy consumption has been shown to be in a linear relationship with the number of generated messages [6], the savings can be calculated as the percent decrease in the number of transmitted messages of our solution as compared to the baseline approach without CUPUS:

$$Savings = \frac{M_{alternative} - M_{CUPUS}}{M_{alternative}} \quad (11)$$

In the following section, we use a real data set to verify our assumptions regarding user distribution, and further provide an analysis of savings for different scenarios using simulations.

V. Model evaluation

A. *Real data set analysis of user distribution in urban areas*

To investigate our assumptions regarding user distribution in urban areas we analyzed the data set presented in [19]. The authors deployed a crowdsensing system in Seoul, South Korea from March 2011 to September 2012, where 85 participants during the campaign actively used the application for 79 days in average. A user smartphone application gathered data from the cellular network provider, GPS sensor, wireless module, microphone and camera which later was used for autonomous place detection. All users generated readings in 13500 distinct places. We note that the dataset does not include information with regards to energy consumption.

First, we pre-filtered the data set to remove faulty entries, such as for example user check-ins without exact location or timestamp. After filtering, we obtained a data set with 151649 user checkins with exact location and timestamp collected from 67 unique users. Since the number of unique users was small, we split the set based on a user identifier and date. Each slice (i.e. a set with user check-ins in a single day) represented an individual user movement pattern during one day. For our analysis, a user-day set is considered as an individual user. The user application, due to energy-savings, did not record user location with high frequency, instead it relied on techniques of user motion detection and sampling strategies to detect change of user location. To obtain data regarding user movement between two sequential user check-ins in a data set, we created additional check-ins as interpolation of user movement between two locations. Additional check-ins were created with a sampling frequency of 1 and 5 minutes.

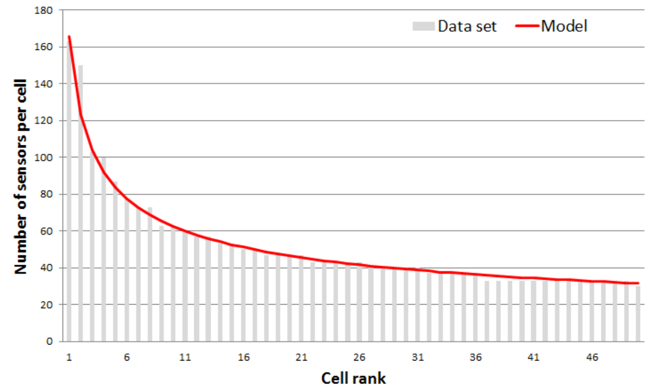

Fig. 5. User distribution with large number of users and frequent check-ins

The analysis of the user distribution in urban areas is highly dependent on the time slots in which an analysis is performed, and the geographical grouping of users in cells which represent a single location context. For geographical grouping, i.e. determining unique cells over wide urban area, we used the military grid reference system (MGRS) [22]. The MGRS is a reference system for locating points on the earth. It allows precision from 100 kilometer up to 1 meter, so it is suitable to be used in urban or rural areas. In our analysis we used a precision of 1 kilometer, i.e. we grouped users based on 1 kilometer squared cells. Due to the limited number of users in the data set and the size of the Seoul area (approx. 600 km^2) we found that 1 square kilometer cells are appropriate to observe grouping of users in such a large area while retaining location connectivity of grouped users. A second important observation criterion is the length of a time period in which a user distribution is observed. Long time periods whose duration is in days or longer are acceptable for cumulative statistics to obtain data about popular places or general user movement patterns. Distribution of users in urban areas is volatile during a single day, each period of a day has its own characteristics (e.g. users in the morning travel to work in business part of a city, and during the afternoon users are dispersed through the city going back home or running personal errands). We therefore analyzed user distribution during the course of a day by using different observation time windows whose range varied from 15 minutes to 24 hours.

Our analysis showed that user distribution in the data set is in accordance with the power law distribution no matter which observation time window is used. Distribution parameters changed for each observation time window, which is expected since users often change location and context in urban areas. Figure 5 depicts the distribution of users observed from 8:30 pm until 9 pm. We analyzed data of 12000 users, all taken randomly from the prepared data set and check-in frequency was once in a minute for all users. The mean absolute error (MAE) was around 0.57 for $\alpha = 0.423$ and $\beta = 165$. Similar results, and a good fitting to a power law distribution was found when the number of users was lower and check-in frequency was once in five minutes. Figure 6 shows a power law distribution fitting when 1400 users was observed

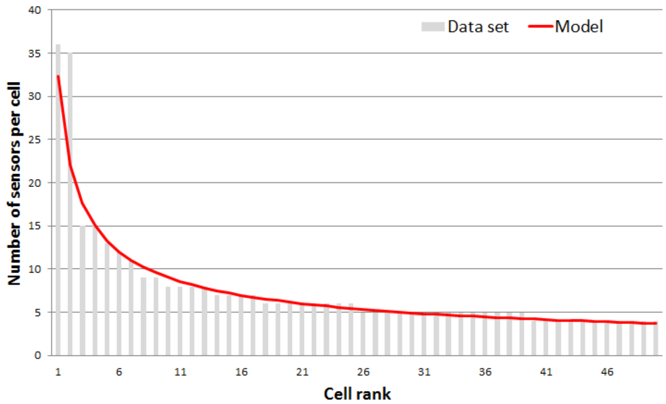

Fig. 6. User distribution with smaller number of users

TABLE I. DEFAULT PARAMETER VALUES

Symbol	Value
N	1000
C	1000
k	5
p_{cov}	0.2
p_{cng}	0.05
m_p	12

between 11 pm and 12 pm. The MAE parameter was 0.22 which represents good fitting results. The β parameter has a value of 35, while the α parameter has a value of 0.553. The α parameter defines the curve slope and hence the sensor distributions across cells, and will be linked to the total energy savings in the next subsection. The β parameter is highly dependent on the number of users and does not affect the shape of the savings curve, it only adjusts its position.

B. Evaluation of CUPUS middleware with QSMF

Following the analysis of empirical data reported in the previous section, in this section we present the evaluation of the CUPUS middleware with QSMF. Hereafter, referring to (11), we analyze the number of transmitted messages in our approach when compared to the alternative approach. Table I shows the default parameter values used in the analysis. We analyze the influence of parameters k, N, p_{cng}, p_{cov} and α on the percent decrease $Savings$. For each analysis, we modified α in combination with one of those 4 parameters, while all other parameters are fixed to default values given in Table I. We note that the default values chosen for p_{cng} and p_{cov} are hypothetical. Our goal is to observe trends in the energy savings with respect to modified influence parameters. Actual values for these parameters in a real world case would be dependent on factors such as the configuration of the city, population, and user mobility.

Since in our model we assume that sensors are located in cells according to a power law distribution, we analyze how the parameter α influences the percent decrease since it models the number of sensors in the most populated cell as well as the distribution tail. Therefore, in all performed analysis we have changed α in the range between 0.2 and 1.4, where a value 0.2 of α indicates a small slope and the value 1.4 indicates a large slope and long tail, as shown in Figure 7.

Figure 8(a) shows how the required number of sensor

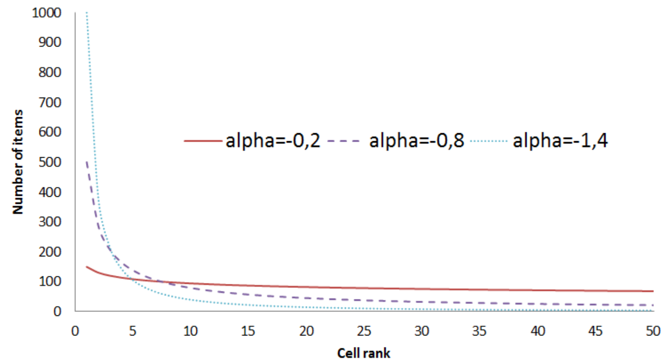

Fig. 7. Example power law distribution for different values of parameter α.

publishers per cell (i.e. k) influences the savings when α changes. As we can see, the percent decrease falls with higher values of k. As expected, the advantage of our approach drops when increasing the value of parameter k since we increase the required number of published messages per cell. Obviously, if more sensors are required per cell, sensor deactivation and data filtering on mobile phones has less influence since we need to satisfy application requirements. Interestingly, we observe that the energy savings curve exhibits a local minimum, after which there is again an increase in savings. Analytical results portray that our approach gives significant savings when α is in the range from 0 to 0.5 as well as from 1.0 to 1.4, but drops when α reaches approximately 0.8. We attribute the shape of the savings curve to the nature of the power law distribution as portrayed in Figure 7, whereby moving towards a larger value of α the total number of sensors falling under the k line increases (more sensors are located in cells that are not covered by subscriptions). However, due to the shape of the curve (long tail and larger slope) and the fact that the percentage of covered cells is kept constant, after a certain value of α we again see an increase in the savings.

Figure 8(b) shows how the percent energy savings increases with an increasing number of sensors N in the system. By increasing the value of parameter N, we increase the number of possible publishers in the system. Therefore, as we expected, the number of sensors located in cells for which there is interest from subscribers (i.e., cells that are covered by subscriptions) is grater than the required number of sensors per cell (i.e. k). This produces savings in the number of published messages since we deactivate $n(c_i) - k$ sensors in a cell c_i for which $n(c_i) > k$. Once again, we observe the lowest savings when α is approximately 0.8.

Figure 8(c) shows how the percent of savings changes with increasing percentage of cells covered by subscriptions p_{cov}. As we can see, the percentage of energy savings falls with p_{cov}. By increasing the value of parameter p_{cov}, we increase the number of cells covered with at least one subscription. As expected, the advantage of our approach drops when increasing p_{cov} due to the drop in retained publications. Obviously, if all cells are covered by subscriptions, there is no value in data filtering on mobile phones as announce and control messages represent an overhead: Our approach drops to 0.1 when p_{cov} reaches 0.8, but it can cause significant savings when p_{cov} is in the range from 0 to 0.5.

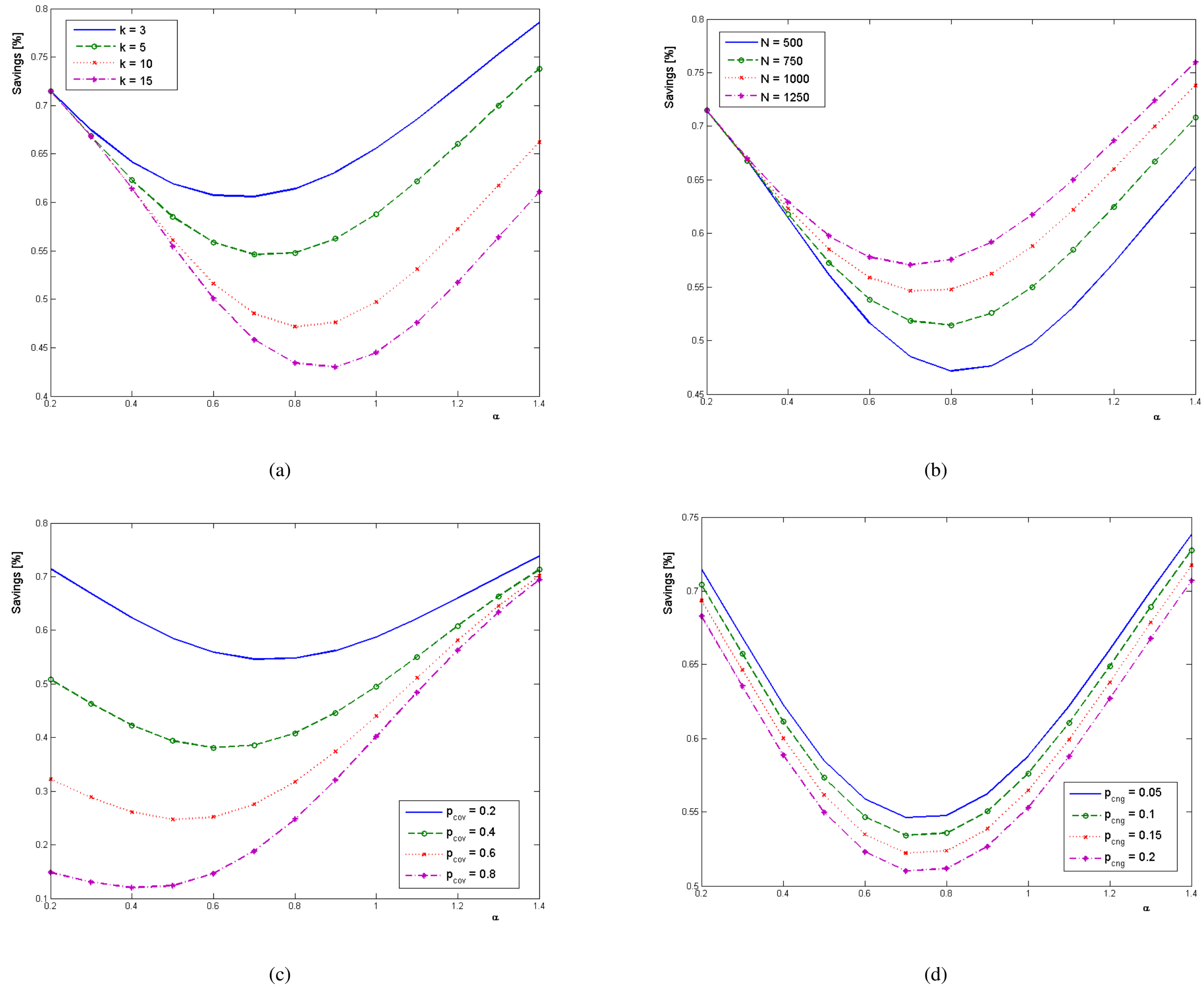

Fig. 8. Energy savings as a function of α for: (a) different application requirements in terms of necessary sensors k; (b) total number of sensors; (c) percentage of cells covered with subscriptions; and (d) percentages of mobile sensors changing their cell

Finally, Figure 8(d) shows how the energy savings changes when increasing the probability that a sensor will go to a neighboring cell between any two consecutive steps. As we can see in the figure, the percentage savings drops with an increase in p_{cng}. By increasing the value of parameter p_{cng} we model the mobility of sensors. Since our approach generates additional *announce* messages when publishers are changing cells as well as additional control messages for activation/deactivation of a sensor in a new cell, obviously the advantage of our approach drops when increasing p_{cng}. As stated previously, our goal is to show energy savings trends. With respect to realistic p_{cng} values for a chosen time interval and context, additional empirical data is needed. Studies addressing such real data are being addressed in our ongoing work.

VI. Conclusions

In this paper, we have focused on mobile crowdsensing applications whereby mobile devices and sensors are used to share data at a community level for the purpose of measuring phenomena of common interest. We have presented our cloud-based publish/subscribe middleware (CUPUS) as a solution developed within the scope of the EU FP7 OpenIoT project and supporting real-time acquisition of sensor data on mobile devices, continuous data processing in the cloud, and near real-time delivery of sensor data to mobile devices. In this work we have particularly focused on energy-efficient and quality-driven sensor management, supported in our system with the addition of a QoS Sensor Management Function, designed to obviate redundant sensor activity and consequently reduce overall system energy consumption. We model energy savings by comparing the calculated number of messages exchanged in our system (as a consequence of sensor mobility and management) with the number of messages that would be exchanged in a system assuming the publication of all collected data.

Following the analysis of a real data set measuring user distribution in a large urban area (Seoul, South Korea), we confirm our assumptions with regards to user distributions as conforming to a power law distribution. We then use simulation results to evaluate savings calculated using a proposed analytical model for different application requirements and geographical sensor distribution scenarios. Results show the possibility of savings ranging from 40% to 80%. We observe that as a consequence of the power law distribution and simulation parameters, the shape of the savings curve as a function of α (determining the rate of the distribution tail decay) exhibits a local minimum at approximately a value of 0.8.

With respect to ongoing work, the CUPUS architecture together with the QSMF have been deployed in the context of an Urban Crowdsensing case study focused on opportunistic sensing of air quality via mobile sensors and devices. The architecture has been integrated with the open source IoT platform developed within the scope of the OpenIoT project, focused on enabling the semantic interoperability of IoT services in the cloud. An ongoing field trial is being run to test system functionality and evaluate performance, involving individuals carrying wearable air quality sensors that continuously contribute sensed data via a smartphone to the cloud platform, making data available to others citizens in the form of a real-time air quality monitoring application. Subsequent data analysis will be conducted to empirically determine sensor distribution and energy savings resulting from the deployed system.

ACKNOWLEDGEMENT

This work has been partially carried out in the scope of the project ICT OpenIoT Project FP7-ICT-2011-7-287305 co-funded by the European Commission under FP7 program.

REFERENCES

[1] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *Comm. Mag.*, vol. 48, no. 9, pp. 140–150, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2010.5560598

[2] L. Atzori, A. Iera, and G. Morabito, "From "Smart Objects" to "Social Objects": The Next Evolutionary Step of the Internet of Things," *Communications Magazine, IEEE*, vol. 52, no. 1, pp. 97–105, 2014.

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

[4] J. A. Burke *et al.*, "Participatory sensing," in *Proc. of WSW'06 at ACM SenSys 2006*. ACM, 2006.

[5] S. Sagiroglu and D. Sinanc, "Big data: A review," in *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, 2013, pp. 42–47.

[6] I. Podnar Zarko, A. Antonic, and K. Pripužic, "Publish/subscribe middleware for energy-efficient mobile crowdsensing," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, ser. UbiComp '13 Adjunct. New York, NY, USA: ACM, 2013, pp. 1099–1110. [Online]. Available: http://doi.acm.org/10.1145/2494091.2499577

[7] A. Antonic, K. Rozankovic, M. Marjanovic, K. Pripuzic, and I. Podnar Zarko, "A mobile crowdsensing ecosystem enabled by a cloud-based publish/subscribe middleware," in *Proc. of FiCloud-2014, to appear*, Aug. 2014.

[8] B. Milosevic, J. Yang, N. Verma, S. S. Tilak, P. Zappi, E. Farella, L. Benini, and T. Simunic Rosing, "Efficient energy management and data recovery in sensor networks using latent variables based tensor factorization," in *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems*. ACM, 2013, pp. 247–254.

[9] M. Riahi, T. G. Papaioannou, I. Trummer, and K. Aberer, "Utility-driven data acquisition in participatory sensing," in *Proceedings of the 16th International Conference on Extending Database Technology*. ACM, 2013, pp. 251–262.

[10] P. P. Jayaraman, C. Perera, D. Georgakopoulos, and A. Zaslavsky, "Efficient opportunistic sensing using mobile collaborative platform mosden," in *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on*. IEEE, 2013, pp. 77–86.

[11] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," *Internet Computing, IEEE*, vol. 16, no. 5, pp. 36–44, 2012.

[12] N. Nikzad, J. Yang, P. Zappi, T. S. Rosing, and D. Krishnaswamy, "Model-driven adaptive wireless sensing for environmental healthcare feedback systems," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3439–3444.

[13] N. Nikzad, N. Verma, C. Ziftci, E. Bales, N. Quick, P. Zappi, K. Patrick, S. Dasgupta, I. Krueger, T. Š. Rosing *et al.*, "Citisense: improving geospatial environmental assessment of air quality using a wireless personal exposure monitoring system," in *Proceedings of the conference on Wireless Health*. ACM, 2012, p. 11.

[14] W. Willett, P. Aoki, N. Kumar, S. Subramanian, and A. Woodruff, "Common sense community: scaffolding mobile sensing and analysis for novice users," in *Pervasive Computing*. Springer, 2010, pp. 301–318.

[15] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, "Approximate data collection in sensor networks using probabilistic models," in *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*. IEEE, 2006, pp. 48–48.

[16] J. Yang, S. Tilak, and T. S. Rosing, "An interactive context-aware power management technique for optimizing sensor network lifetime."

[17] C. Perera, P. P. Jayaraman, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensor discovery and configuration framework for the internet of things paradigm," *arXiv preprint arXiv:1312.6721*, 2013.

[18] G. Mühl, L. Fiege, and P. Pietzuch, *Distributed event-based systems*. Springer, 2006, vol. 1.

[19] Y. Chon, N. D. Lane, Y. Kim, F. Zhao, and H. Cha, "Understanding the coverage and scalability of place-centric crowdsensing," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 2013, pp. 3–12.

[20] A. Noulas, S. Scellato, R. Lambiotte, M. Pontil, and C. Mascolo, "A tale of many cities: universal patterns in human urban mobility," *PloS one*, vol. 7, no. 5, p. e37027, 2012.

[21] K. Schnaitter and N. Polyzotis, "A benchmark for online index selection," in *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*. IEEE, 2009, pp. 1701–1708.

[22] T. D'Roza and G. Bilchev, "An overview of location-based services," *BT Technology Journal*, vol. 21, no. 1, pp. 20–27, 2003.