ecoSense: Minimize Participants' Total 3G Data Cost in Mobile Crowdsensing Using Opportunistic Relays

Leye Wang, Daqing Zhang, *Member, IEEE*, Haoyi Xiong, *Member, IEEE*, John Paul Gibson, Chao Chen, and Bing Xie

***Abstract*—In mobile crowdsensing (MCS), one of the participants' main concerns is the cost for 3G data usage, which affects their willingness to participate in a crowdsensing task. In this paper, we present the design and implementation of an MCS data uploading mechanism—ecoSense—to help reduce additional 3G data cost incurred by the whole crowd of sensing participants. By considering the two most common real-life 3G price plans—unlimited data plan (UnDP) and pay as you go (PAYG), ecoSense partitions all the users into two groups corresponding to these two price plans at the beginning of each month, with the objective of minimizing the total refunding budget for all participants. The partitioning is based on predicting users' mobility patterns and sensed data size. The ecoSense mechanism is designed inspired by the observation that during the data uploading cycles, UnDP users could opportunistically relay PAYG users' data to the crowdsensing server without extra 3G cost, provided the two types of users are able to "meet" on a common local cost-free network (e.g., Bluetooth or WiFi direct). We conduct our experiments using both the Massachusetts Institute of Technology reality mining and the Small World In Motion (SWIM) simulation data sets. Evaluation results show that ecoSense could reduce total 3G data cost by up to ∼50%, when compared to the direct-assignment method that assigns each participant to UnDP or PAYG directly according to the size of her sensed data.**

***Index Terms*—3G data cost, data relay, delay-tolerant data uploading, incentive, mobile crowdsensing (MCS).**

Manuscript received July 20, 2015; accepted January 3, 2016. Date of publication February 19, 2016; date of current version May 15, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61572048, in part by the Microsoft Collaboration Research Project, in part by the EU FP7 MONICA Project under Grant PIRSES-GA-2011-295222, in part by the Open Research Fund Program of the Shenzhen Key Laboratory of Spatial Smart Sensing and Services, Shenzhen University, and in part by the Chongqing Basic and Frontier Research Program under Grant cstc2015jcyjA00016. This paper was recommended by Associate Editor J. Lu.

L. Wang is with SAMOVAR, Télécom SudParis, Centre National de la Recherche Scientifique, Université Paris-Saclay, Evry 91011, France (e-mail: leye.wang@telecom-sudparis.eu).

D. Zhang is with SAMOVAR, Télécom SudParis, Centre National de la Recherche Scientifique, Université Paris-Saclay, Evry 91011, France, and also with the Key Laboratory of High Confidence Software Technologies, Ministry of Education, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: daqing.zhang@telecom-sudparis.eu).

H. Xiong is with the University of Virginia, Charlottesville, VA 22904-4747 USA (e-mail: haoyi.xiong.fr@ieee.org).

J. P. Gibson is with Télécom SudParis, Evry 91011, France (email: paul.gibson@telecom-sudparis.eu).

C. Chen is with Chongqing University, Chongqing 400044, China (e-mail: cschaochen@cqu.edu.cn).

B. Xie is with the Key Laboratory of High Confidence Software Technologies, Ministry of Education, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: xiebing@sei.pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2016.2523902

I. Introduction

NOWADAYS, mobile crowdsensing (MCS) [1], [2] is becoming an effective and practical way to carry out various sensing tasks, as rich-sensor equipped smartphones are getting more and more popular. However, some obstacles severely stop users from participating in MCS tasks. For example, participating in MCS tasks will incur additional 3G data usage for a user, which might lead to more fees paid to the telecom operators. This issue, which we will refer to as 3G data cost, is one of the major concerns for MCS participants [3], [4].

To deal with this issue, some existing MCS projects ask users to delay uploading sensed data until they have a WiFi connection; apparently this method might lead to long uncontrollable delay between data sensing and uploading. Such a long uncontrollable delay might harm many MCS tasks. On one hand, uncontrollable delay might not be tolerable for most real-time monitoring applications, as the delayed sensed data becomes valueless; even if an MCS task allows some delay, uncontrollable delay may still exceed the maximum tolerable delay. On the other hand, long uncontrollable delay increases the probability of sensed data loss due to abnormal events on participants' mobile phones (e.g., running out of storage space).

Another effective way to mitigate participants' 3G data cost concern is providing participants with incentives to cover any additional 3G data cost arising from participation in the MCS task. Compared to WiFi-only uploading, refunding 3G data cost encourages participants to willingly upload data at any specific time via 3G, which can reduce and control the delay between data sensing and uploading. However, 3G data refund increases an organizer's total MCS task budget. For MCS tasks which need a big number of participants, in particular, this 3G data refund budget can be significant. Thus, "how to reduce the 3G data refund budget" becomes a critical problem for MCS organizers.

In this paper, we try to address the above problem. First, we study the common price plans of 3G data cost. Currently, two price plans are widely used by most telecom operators: 1) unlimited data plan (UnDP) and 2) pay as you go (PAYG).

1) *Unlimited Data Plan:* With UnDP, a user can transfer an unlimited amount of data during a period of time (usually for a month). The cost for an UnDP is fixed, e.g., \$7/month (denoted as Price_u).
2) *Pay As You Go:* With PAYG, a user pays 3G data cost according to the amount of data transferred via 3G, e.g., \$0.1/MB (denoted as Price_p).

With the above two 3G price plans, a simple solution to refunding participants' 3G data cost is to choose the right refund scheme for each mobile user according to the amount of her uploaded data. Specifically, this direct-assignment method works as follows.

1) For each participant u_i, estimate her amount of sensed data to be uploaded each month (d MB).
2) Two possible refund schemes exist.
   a) *UnDP:* Refund is Price_u.
   b) *PAYG:* Refund is $d * \text{Price}_p$.

   Choose the cheaper one as the refund for u_i.
3) If assigned to UnDP, u_i needs to set her 3G price plan to UnDP before the next month starts (this is why we need to estimate d). At the end of the next month, the organizer pays Price_u to u_i.
4) If assigned to PAYG, u_i can keep her original personal 3G price plan for next month (independent of whether u_i's original price plan is PAYG or UnDP, the organizer does not need to know).[1] In next month, the organizer counts the actual amount of sensed data that u_i uploads (d' MB). At the end of the next month, the organizer pays $d' * \text{Price}_p$ to u_i.

When the sensing task is determined, one participant's sensed data size in a month can usually be estimated within reasonable error bounds (i.e., $d \approx d'$), which makes direct-assignment applicable.

Although direct-assignment can support real-time data uploading reasonably well, for many MCS tasks which do not require real-time uploading (i.e., allowing a max tolerable delay between sensing and uploading), the refund budget of direct-assignment may be very high. The following events can be leveraged to reduce participants' 3G data cost during the delay period, so that the organizer's refund budget can be reduced.

1) A PAYG participant can use a cost-free network, such as Bluetooth or WiFi (e.g., at home or in the office), to upload sensed data to the server within the delay period, which reduces her 3G data cost.
2) UnDP participants can help relay PAYG participants' sensed data to the server. This kind of relay reduces 3G data cost for PAYG participants, without increasing 3G data cost for UnDP participants, thus decreasing the organizer's refund budget.

Based on these events, in this paper, we design a novel data uploading framework for MCS, called ecoSense, whose goal is to optimally partition the participants into UnDP or PAYG subgroups, in order to minimize the organizer's 3G data refund budget, via maximumly taking advantage of cost-free networks and mobile participants as data relays.

Two important issues are involved in designing ecoSense.

A. How to Transfer Data When Two PAYG Participants Meet

Unlike the clear relay strategy between a PAYG participant and an UnDP participant (i.e., PAYG $\xrightarrow{\text{data}}$ UnDP), relay strategy between two PAYG participants is more complicated and will affect the organizer's refund budget. Among all the possible strategies, flooding (i.e., always exchanging data between two PAYG participants) is expected to produce the smallest refund budget. Because after flooding, if any one of two PAYG participants could meet an UnDP participant or a cost-free network, both of their data could be uploaded without 3G cost. However, flooding might incur too many redundant relays that rapidly drain the batteries of participants' mobile phones. Though this paper focuses on minimizing the organizer's 3G data refund budget, the participants' energy concerns should also be taken into account to some extent. Otherwise, even if ecoSense "successfully" minimizes the refund budget, participant phones' energy consumption might be too high, making ecoSense impractical. Thus, to study the tradeoff between the organizer's 3G data refund budget and participant phones' energy consumption, we try different data uploading/relay strategies following the state-of-the-art from our previous work [5], [6].

B. How to Decide Each Participant's Refund Scheme—PAYG or UnDP

To minimize the organizer's 3G data refund budget, another key issue is to determine which participants should be assigned to each of the schemes (and not just the percentage of participants allocated to each scheme). Thus, the participant partition algorithm needs to consider each participant's mobility pattern and sensed data size.

1) *Mobility Pattern:* To maximize data relay opportunities between PAYG and UnDP participants, accurately profiling each participant's mobility pattern is necessary for deciding whether she should be assigned to the UnDP or PAYG scheme. Intuitively, "active" participants who can help more other participants relay data should be assigned to UnDP.
2) *Sensed Data Size:* A participant's sensed data size would also impact whether she is assigned to PAYG or UnDP. Generally, a participant who uploads a larger amount of sensed data should be assigned to UnDP.

Thus, our proposed partition algorithm first predicts each participant's mobility pattern and estimates her sensed data size, and finally uses a genetic algorithm to obtain a participant partition for PAYG and UnDP groups. This algorithm is run before a new month starts (only once a month) on the MCS organizer's server.[2]

[1] A participant's refund scheme may be different from her 3G price plan. Refer to Sections VIII-C and VIII-D for more details. In rest of this paper, unless specified otherwise, UnDP and PAYG represent refund schemes.

[2] Most telecom operators need a participant to decide her 3G data price plan for the upcoming month before the end of the current month.

In summary, this paper makes the following contributions.

1) To the best of our knowledge, this is the first work that aims to minimize the organizer's 3G data refund budget by leveraging heterogeneous networks (e.g., 3G, Bluetooth, and WiFi) and delay-tolerant data uploading mechanisms in MCS.
2) We propose an MCS data uploading framework, called ecoSense, attempting to minimize the organizer's 3G data refund budget. ecoSense considers two 3G price plans to refund the participants—PAYG and UnDP—and proposes data uploading strategies for both UnDP and PAYG participants in the delayed uploading period. Furthermore, a participant partition algorithm is designed to split all the participants between PAYG/UnDP participant groups, in order to minimize the organizer's 3G data refund budget, via maximumly taking advantage of opportunistically encountered cost-free networks and mobile participants as data relays.
3) We use a real-life data set, i.e., the Massachusetts Institute of Technology (MIT) reality mining [7], and a larger Small World In Motion (SWIM) [8] simulation data set to evaluate our approach. The evaluation results show that ecoSense can reduce the organizer's 3G data refund budget by up to ~50% compared to the direct-assignment solution.

II. Related Work

Recent years have witnessed an increasing interest in MCS research [1], [2], leading to many applications, such as urban noise monitoring [9], everyday point-of-interest tracking [10], and social interaction sensing [11]. Generic MCS platforms, such as ParticipAct [12], have also been proposed to support various crowdsensing campaigns. As successful MCS tasks usually require a large number of participants, the question of how to incentivize users to participate is a key issue for MCS organizers. Potential participants need to have their concerns addressed, and current participants must not be discouraged by any inconvenience arising from the sensing process.

Previous research work about MCS incentives has leveraged game theory and auction mechanisms to analyze the optimal payment to be offered by the MCS organizer to participants and to find the best compromise between participants' and organizer's profit (i.e., the utility function in game theory) [13]. Some other work attempts to reduce the incentives by minimizing the number of recruited participants [14], [15] or amount of collected data [16]–[18], while ensuring a certain level of the task quality. As an alternative to monetary reward, some approaches offer other incentives such as service time [3] and coupons [19]. In general, these approaches assume the users' cost to finish a task to be known in advance, and this cost follows some specific probability distribution in their simulation experiments. In contrast, our approach analyzes users' cost from a more pragmatic viewpoint—we focus on the 3G data cost, which has previously been shown to be a main concern for a majority of users [3], [4].

With respect to the objective of reducing 3G data cost, previous research work proposes to reduce the sensed data size based on mechanisms such as: 1) compression/aggregation of data on the phone [11], [20] and 2) uploading only a subset of the data while deducing the rest [21]. As ecoSense is a data uploading framework, all the aforementioned mechanisms, which can reduce sensed data size before uploading, could be incorporated into ecoSense to reduce users' 3G data cost further. Cost-free data uploading methods for MCS applications, such as user node relays [22] and delayed WiFi/Bluetooth transfers [5], [6], [23], have already been proposed in existing research work; ecoSense uses similar mechanisms. However, by considering two common 3G data price plans—UnDP and PAYG—we identify a novel problem to minimize the MCS organizer's 3G data refund budget, through optimally partitioning the users to PAYG/UnDP groups and designing uploading strategies for users, so that UnDP users can help relay PAYG users' data efficiently.

Existing work concerning human mobility pattern prediction is also relevant, since ecoSense predicts users' mobility patterns in order to determine the participant partition. Human mobility pattern prediction is an increasingly important research area, with many open questions yet to be resolved. For example, most existing research studies only the short-time "next place" prediction [24], [25], which would not necessarily be appropriate for the longer-term mobility prediction that is required by ecoSense (i.e., users' mobility for the whole upcoming month). As this paper does not focus on designing such a long-term mobility prediction algorithm, we currently use a state-of-the-art mobility prediction method based on Poisson distribution (Section VI-A), which is widely applied in previous work, e.g., [6], [15], [16], [26]–[28].

Although ecoSense and our previous work effSense [5], [6] both leverage heterogeneous networks to save 3G cost in delay-tolerant MCS, the research assumptions and problems are distinct from each other. In effSense, we predefine whether a participant has UnDP and focus on designing data uploading strategies; while in ecoSense, in addition to uploading strategies, the key technical issue that needs to be addressed is deciding whether to assign a participant to the UnDP or PAYG group.

III. Problem Statement

In this section, we first introduce the MCS task process that ecoSense can be applied to, i.e., delay-tolerant MCS. Afterward, we formulate the research problem of ecoSense formally.

A. Delay-Tolerant MCS

Many crowdsensing tasks (e.g., MIT reality mining [7] and environment monitoring [15], [16], [27], [28]) do not require immediate uploading of the data after it is sensed (called delay-tolerant MCS task). Such tasks allow some delay (max tolerable delay T_d) between collecting the data from sensors and uploading it to the server, i.e., the sensed data generated at t on a participant's phone can be uploaded during $[t, t+T_d]$.

Formally, in this paper, we consider a crowdsensing task process that is composed of two kinds of cycles: 1) sensing cycles and 2) delayed-uploading cycles (see Fig. 1).

1) *Sensing Cycle:* A crowdsensing task process can be split into continuous sensing cycles. As shown in Fig. 1, each

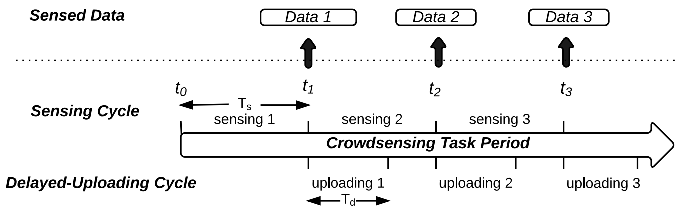


Fig. 1. Sensing cycles and delayed-uploading cycles.

sensing cycle lasts for T_s, i.e., the ith sensing cycle starts at $t_{i-1} = t_0 + (i-1)T_s$ and ends at $t_i = t_0 + iT_s$. We assume that each participant's sensed data are prepared for uploading right up until the end of each sensing cycle (e.g., some aggregation algorithms need to be run on the raw sensed data before uploading).

2) *Delayed-Uploading Cycle:* The ith delayed-uploading cycle starts at the end of the ith sensing cycle (i.e., t_i) and lasts for T_d (i.e., ends at $t_i + T_d$). In the ith delayed-uploading cycle, each participant attempts to upload her ith sensing cycle's data to the server. Various data uploading/relay strategies could be applied here (e.g., flooding, spray-and-wait [29], and the strategies in our previous work [6]). At the end of a delayed-uploading cycle, if some participants' data are still not uploaded to the server, these participants are forced to upload their data to the server via 3G, in order to ensure that all the sensed data could arrive at the server within the delay T_d. In this paper, we consider only $t_i + T_d \leq t_{i+1}$ (i.e., $T_d \leq T_s$), which means that participants need to upload their ith sensing cycle's data to the server before their $i+1$th sensing cycle's data gets ready (so different delayed-uploading cycles will not overlap).

In summary, ecoSense can be applied to any MCS task process that meets two requirements: 1) all the participants' sensed data is ready for uploading at the end of each sensing cycle and 2) the ith sensing cycle's data needs to be uploaded to the server before the $i+1$th sensing cycle's data gets ready. Therefore, in each delayed-uploading cycle, a participant only needs to upload one piece of sensed data.

B. Problem Formulation

Considering the above MCS task process, ecoSense aims to reduce the crowdsensing organizer's refund budget for participants' 3G data cost in delayed-uploading cycles. Before formulating the problem formally, we introduce some key concepts.

Definition 1 (Cost-Free Events): A cost-free event refers to an encounter between a PAYG participant and another participant or device that can probably help PAYG participants upload data to the server without 3G data cost.[3]

Definition 2 (Uploading Decision Making): In a delayed-uploading cycle with maximum tolerable delay T_d, when a PAYG participant u_i with sensed data r_i (generated at t) encounters a cost-free event e at time t^* ($t^* \in [t, t+T_d]$), ecoSense makes a decision about whether data r_i needs to be uploaded/relayed (i.e., true) or not (i.e., false). We express this decision function as $\mathcal{D}(u_i, r_i, t^*, e, t, T_d) \rightarrow \{\text{true, false}\}$.

In this paper, if the decision $\mathcal{D}$ is true, we assume that a PAYG participant can always relay/upload data successfully.

Definition 3 (Participant Partition): Given all the crowdsensing participants U, a participant partition assigns each participant to UnDP group (U_u) or PAYG group (U_p). We express this partition function as $\mathcal{P}(U) \rightarrow [U_u, U_p]$, where $U_u \cup U_p = U$ and $U_u \cap U_p = \phi$.

Based on these definitions, we formulate our problem as follows.

1) Problem Formulation: In a crowdsensing task allowing certain data uploading delay (T_d), given all the participants (U), unit prices for both 3G price plans UnDP (Price_u, e.g., \$7/user for a month) and PAYG (Price_p, e.g., \$0.1/MB), we require an uploading decision-making strategy for PAYG participants ($\mathcal{D}$) and a PAYG/UnDP participant partition function ($\mathcal{P}$), in order to minimize the crowdsensing organizer's refund budget for all the participants' additional 3G data cost in one month[4]

$$\begin{aligned}\underset{\mathcal{D},\mathcal{P}}{\operatorname{argmin}}\ \text{Refund} &= \underset{\mathcal{D},\mathcal{P}}{\operatorname{argmin}} \left(\text{Refund}_u + \text{Refund}_p\right)\\ &= \underset{\mathcal{D},\mathcal{P}}{\operatorname{argmin}} \left(|U_u| * \text{Price}_u + \sum_{i \in U_p} d_i * \text{Price}_p\right)\end{aligned}$$

where

1) *Refund_u:* refund budget for UnDP participants;
2) *Refund_p:* refund budget for PAYG participants;
3) *d_i:* amount of data uploaded via 3G by participant i in a month.

The solution to this problem is nontrivial, because:

1) we can neither foresee the participants' mobility traces in the next month, nor how much sensed data that needs to be uploaded. Thus, obtaining d_i is not straightforward: both participant mobility and sensed data size prediction methods should be combined in order to estimate d_i;
2) different $\mathcal{D}$ and $\mathcal{P}$ would affect Refund_u and Refund_p jointly. For example, if $\mathcal{P}$ assigns more participants to UnDP, then Refund_u increases and Refund_p decreases, so that whether overall Refund increases or decreases remains uncertain. Even if $\mathcal{P}$ is determined, $\mathcal{D}$ can still impact Refund_p, because PAYG participants hold different uploading strategies under different $\mathcal{D}$.

IV. Overview of ecoSense

To solve the problem formulated in the previous section, we design a novel MCS data uploading framework named ecoSense. In this section, we first use a running example to illustrate the basic idea of ecoSense and compare it with direct-assignment. Then, we give the overview of our proposed ecoSense framework.

[3] Cost-free events are considered only for PAYG participants, as UnDP participants can upload an unlimited amount of data via 3G.

[4] The problem is parameterized by the frequency with which we can update the participants' type of price plan. Currently, this is usually done on a monthly basis, but our approach is generalizable to different update frequencies which may be more likely in the future provision of crowdsensing services.

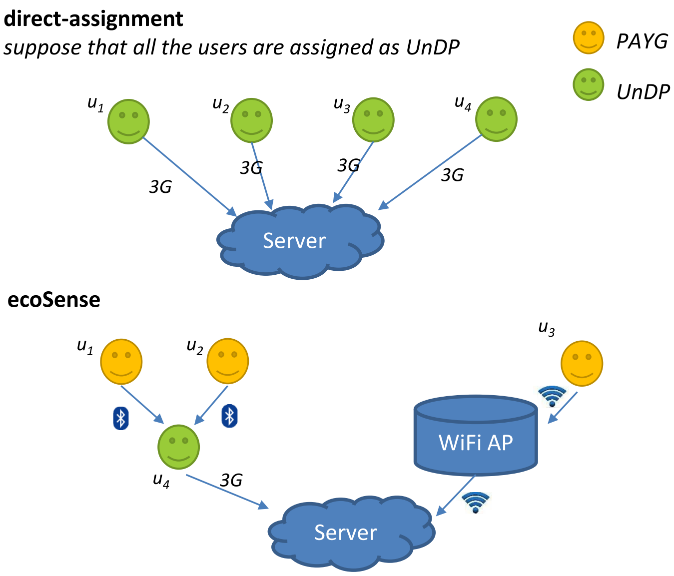


Fig. 2. Comparison between ecoSense and direct-assignment.

A. Running Example: ecoSense Versus Direct-Assignment

To better illustrate the basic idea of ecoSense, we use an example to compare ecoSense and direct-assignment (see Fig. 2). Suppose that Price_u is \$7/month, Price_p is \$0.1/MB, and each participant's sensed data will be larger than 70 MB/month, so direct-assignment refunds each of the four participants as UnDP. The refund budget of direct-assignment is $\$7 * 4 = \$28$ for a month.

By allowing some delay between data sensing and uploading, ecoSense enables the following new data uploading paths.

1) u_1 and u_2 have high probability to encounter u_4 via Bluetooth within the delay period, thus u_4 can relay u_1 and u_2's data.
2) u_3 is likely to meet a free WiFi access point (AP) within the delay period, so u_3 could upload data via that AP without 3G data cost.

As a result, by using delay-tolerant data uploading mechanisms, u_1, u_2, and u_3 could significantly reduce the amount of uploaded data via 3G. By adopting the above mechanisms, assume that u_1, u_2, and u_3's 3G-uploaded sensed data size can decrease to 50, 60, and 40 MB, respectively, then ecoSense would choose the PAYG scheme for the three participants instead of UnDP, reducing the organizer's refund budget to $\$0.1 * (50 + 60 + 40) + \$7 * 1 = \$22$ for a month, compared to the \$28 of direct-assignment.

B. Overview of the ecoSense Framework

The overview of our ecoSense framework is shown in Fig. 3, which contains two key components.

1) Uploading Decision-Making (Client Component): This component runs on every crowdsensing participant's smartphone. It is triggered to decide whether to upload/relay or to keep data when a participant encounters a cost-free event, such as meeting another participant or discovering a Bluetooth/WiFi gateway. The uploading decision-making component will be further elaborated in Section V.

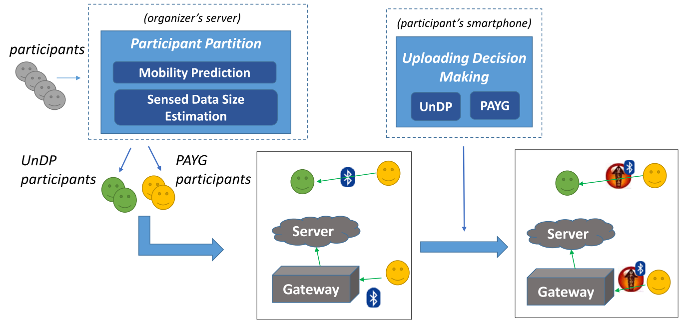


Fig. 3. Overview of the ecoSense framework.

2) Participant Partition (Server Component): This component runs on the crowdsensing organizer's server to assign the participants to either the UnDP or the PAYG group. It relies on two modules—mobility prediction and sensed data size estimation.

1) Mobility prediction module predicts participants' mobility patterns in the next month. With mobility prediction results, we can forecast how often a participant might meet another participant, a Bluetooth gateway, a WiFi AP, etc.
2) Sensed data size estimation module estimates the amount of sensed data that a specific participant would contribute in the following month. For different participants, sensed data size might vary according to their activeness, privacy concerns, visited locations, etc.

Currently, most telecom operators' 3G data plans can change once a month, so this component needs to run once at the end of a month, to obtain the group partition for the following month. The participant partition component will be further elaborated in Section VI.

Now we briefly explain ecoSense's workflow during a crowdsensing task period.

1) As shown in the left part of Fig. 3, before a new month begins, the participant partition component partitions all the participants into two groups with two different 3G refund schemes: a) UnDP and b) PAYG.
2) After the new month starts, in each delayed-uploading cycle, when a participant encounters a cost-free event (e.g., encountering a Bluetooth gateway or another participant), the uploading decision-making component decides whether to upload/relay or to keep data. For example, in the right part of Fig. 3, after making the decision, a PAYG participant relays data to an UnDP participant via Bluetooth, while another PAYG participant relays data to a Bluetooth gateway.
3) At the end of each delayed-uploading cycle, ecoSense checks all the participants to see whether they have nonuploaded data, which can include the sensed data collected by a participant herself and relayed data received from other participants. Then, ecoSense forces those participants with outstanding nonuploaded data to create 3G connections in order to upload it at the end of the cycle. In fact, only under this condition will PAYG participants upload sensed data with 3G data cost in a particular cycle.

V. Uploading Decision-Making

In this section, we focus on introducing the strategies used in the uploading decision-making component (called uploading strategy), especially for PAYG participants, because their uploading strategy will affect the organizer's 3G data refund budget. To make this paper complete, we also introduce the uploading strategy for UnDP participants.

A. PAYG Uploading Strategy

We provide PAYG participants with three candidate uploading strategies: 1) OneRelay; 2) OneHopFlooding; and 3) Epidemic.

1) OneRelay: A PAYG participant u_p would relay her data when she encounters an UnDP participant u_u (or a Bluetooth/WiFi gateway) at the first time in the delayed-uploading cycle. After this relay, u_p's data uploading process ends successfully as u_u (or the gateway) would help upload u_p's data to the server. The name "OneRelay" just means that a PAYG participant will relay her data at most once. Note that using OneRelay, a PAYG participant will not relay her data to another PAYG participant.

2) OneHopFlooding: A PAYG participant u_p would relay her data unconditionally to another PAYG participant encountered until u_p meets either one of the two following stopping criteria: 1) u_p directly encounters an UnDP participant (or a gateway) u_u and relays data to u_u or 2) the server notifies u_p that she could stop flooding (which we will further discuss in the next section—UnDP uploading strategy). The name "OneHopFlooding" just means that the flooding is only one-hop, i.e., a PAYG participant u_{p1} will only flood her own data (i.e., u_{p1}'s data) to the PAYG participants encountered. That is, even if u_{p1} previously received a PAYG participant u_{p2}'s data, u_{p1} will not further flood u_{p2}'s data to other PAYG participants. However, if u_{p1} encounters an UnDP participant (or a gateway) u_u, u_{p1} will relay both u_{p1} and u_{p2}'s data to u_u.

3) Epidemic [30]: Removing the one-hop restriction of OneHopFlooding can lead to a complete flooding strategy, i.e., Epidemic routing. Using the Epidemic strategy, when two PAYG users meet, they will exchange all the data that they do not have in common, independent of whether the data are generated by themselves or received from someone else. To avoid redundant connections, a reconnection time threshold is set to ensure that two specific users exchange data at most once within a predefined time period [30]. For example, supposing the threshold is 30 mins, then if two users u_1 and u_2 exchange data at 12:00, they will not reconnect with each other and exchange data until 12:30, even though they reencounter between 12:00 and 12:30. Although a smaller threshold could incur a lower refund budget, in practice, we cannot greatly reduce the reconnection time threshold due to the energy consumption issue.

In general, among the three strategies, Epidemic incurs the most data relays, while it can help the organizer to pay the smallest refund budget. OneRelay makes every relay valuable, but the refund budget is likely to be higher than for the other two strategies. All the strategies take the energy consumption (i.e., relay count) into account, as we cannot make the delayed uploading process too energy-draining in real-life scenarios. A detailed comparison of the three uploading strategies with respect to refund budget and energy consumption will be examined in our experiments.

Currently, we restrict our comparison to OneRelay, OneHopFlooding, and Epidemic, because they are easily implemented in participants' mobile phones. In our future work, we will evaluate other PAYG uploading strategies, such as binary spray and wait [29] and strategies proposed in our previous work [5], [6].

B. UnDP Uploading Strategy

How UnDP participants upload data during delayed-uploading cycles does not affect the crowdsensing organizer's 3G data refund budget. Thus, in this paper, we simply assume that the UnDP participants use the UpEnd strategy, defined as follows.

1) UpEnd: UnDP participants keep all the sensed data collected by themselves and received from other PAYG participants, until the end of the delayed-uploading cycle; at which point they upload all the data together.

Specifically, if PAYG participants adopt OneHopFlooding or Epidemic, UnDP participants will perform an additional action when they receive PAYG participants' data: if an UnDP participant u_u receives a PAYG participant u_p's data, u_u will notify the server that u_p's data will definitely be uploaded to the server by u_u before the end of the delayed-uploading cycle. Then, the server will notify u_p to stop flooding her data,[5] which corresponds to the second stopping criterion that we described previously in OneHopFlooding (the same stopping criteria also apply to Epidemic). While this additional action would not affect the organizer's 3G data refund budget, it could decrease the relay count of PAYG participants significantly and make OneHopFlooding and Epidemic more energy-efficient in real-life scenarios.

VI. Participant Partition

After choosing the uploading strategy for the participants, the crowdsensing organizer also needs to partition the participants into two groups—PAYG and UnDP—in order to minimize the 3G data cost that needs to be refunded. Fig. 4 shows the overview of the participant partition framework. To achieve a reasonable participant partition, two factors need to be considered.

1) *Mobility Pattern:* A participant's mobility pattern affects how often she could meet another participant or a Bluetooth/WiFi gateway.
2) *Sensed Data Size:* Different participants will most likely contribute different sizes of sensed data due to variant behaviors such as their degree of activity and their privacy concerns.

In this section, we first describe our methods to predict participants' mobility pattern and to estimate participants' sensed

[5] This notification will consume some data cost for PAYG participants. However, the amount of data cost incurred by the notification is usually small compared to the sensed data to upload. So we currently ignore it.

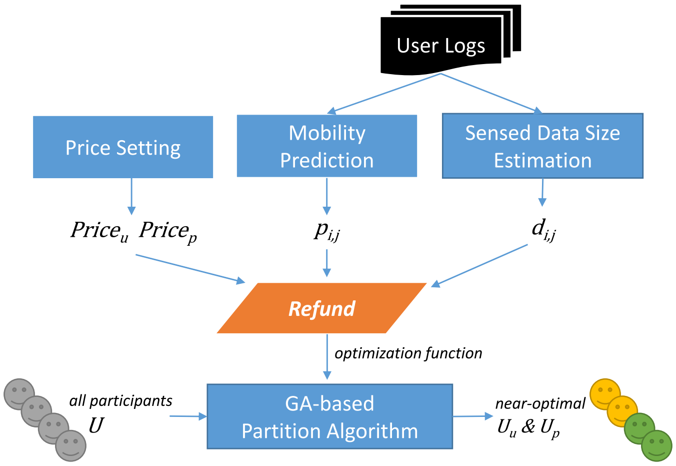


Fig. 4. Overview of participant partition framework.

data size. Then, we propose a genetic algorithm to partition the participants into UnDP and PAYG groups.

A. Mobility Pattern Prediction

Though mobility prediction has been studied comprehensively, most of the existing work focuses on short-term next place prediction [24], [25] and could not be applied directly to ecoSense. In ecoSense, we want to solve a long-term mobility pattern prediction problem.

In the next upcoming month, for each delayed-uploading cycle, what is the probability that a participant encounters another participant or a Bluetooth/WiFi gateway?

Specifically, through previous mobility trace analysis, we predict whether a PAYG participant can upload data without 3G data cost in a delayed-uploading cycle, i.e., the probability that she could meet at least one UnDP participant or one Bluetooth/WiFi gateway. We note this probability as $p_{i,j}$ for participant i and delayed-uploading cycle j.

To predict $p_{i,j}$, we use a Poisson-distribution-based method like [6], [15], [16], [27], and [28].

1) Represent a delayed-uploading cycle j as a triple: $(t_{\text{start}}, t_{\text{end}}, \text{day\_type})$, e.g., (8:00, 12:00, weekday), where t_{start} and t_{end} are the start and end time of the cycle, respectively, and day_type refers to weekday or weekend.
2) Find all the historic time spans with the same triple as the delayed-uploading cycle j (denoting the set of these historic time spans as HS_j), then count the total number of the cost-free events that could help u_i to upload data without additional 3G cost occurred in HS_j, denoted as $\#\text{event}_{\text{free}}(u_i, \text{HS}_j)$. Then $p_{i,j}$ can be predicted as follows, assuming the Poisson distribution stands:

$$p_{i,j} = 1 - e^{\#\text{event}_{\text{free}}(u_i,\text{HS}_j) \big/ |\text{HS}_j|}.$$

A more complicated prediction algorithm might be incorporated into ecoSense, while designing such a prediction method is not the focus of this paper. We note, however, that our experiments demonstrate that using this Poisson-distribution-based prediction method, ecoSense could achieve significant refund savings compared to the direct-assignment method (described in Section I). Furthermore, our experiments show that with this prediction method, ecoSense can achieve a refund budget close to the optimum (foreknowing the participants' mobility traces), which means that such a prediction method is probably good enough for most real-life crowdsensing scenarios.

B. Sensed Data Size Estimation

Estimating how much sensed data each participant needs to upload is also important to obtain an optimal participant partition to minimize total 3G cost. In this section, we attempt to model a participant i's sensed data size during the sensing cycle j, denoted as $d_{i,j}$.

1) Fixed-Size Sensed Data: Some sensing tasks will generate similar sensed data size in a sensing cycle. For example, the size of each user's accelerometer record is approximately proportional to the sensing duration time. So if all the users participate in a common sensing task for the same duration (e.g., activity recognition during daytime) in a sensing cycle, their contributed sensed data size will be similar.

For fixed-size sensing tasks, we model $d_{i,j}$ as

$$d_{i,j} = c$$

where c is a constant, which means that for different participants and different sensing cycles, this sensing task would generate the same sensed data size.

2) Varied-Size Sensed Data: Some sensing tasks will generate different sensed data sizes for different participants in a sensing cycle for various reasons.

1) *Location-Centric Sensing:* This kind of sensing task usually triggers sensing when a participant enters a new location within a target sensing area, e.g., air quality and noise monitoring. Thus, the more places a participant visits, the more sensed data she would gather.
2) *Activeness in Participatory Sensing:* Participatory sensing needs participants to be actively involved in the sensing tasks, e.g., taking photos at specific locations. Different participants can have different levels of activity, leading to different contributed sensed data sizes.
3) *Private Concern:* To protect privacy, some participants might choose not to upload part of the sensed data.

For varied-size sensing tasks, in this paper we consider only the influence of locations visited on the size of the sensed data while ignoring other factors such as activeness and privacy, and model $d_{i,j}$ as

$$d_{i,j} = c + k * l_{i,j}$$

where

1) c is the size of the constant part of the sensed data. Though the data sizes of varied-size sensing tasks usually vary for different participants, still some part of the sensed data size is constant (e.g., some aggregation/summary information of the sensing task). In fact, we can also see fixed-size sensing tasks as a special form of varied-size sensing tasks;
2) $k * l_{i,j}$ is the sensed data size for location-centric sensing, where k is the unit sensed data size for one location,

Algorithm 1 GA-based Partition Algorithm

Input: *Refund*: optimization function; U: all participants
Output: U_p and U_u: participant partition
1: $B \leftarrow bivector(U)$ {Change U into a bi-vector, where each element marks a participant as PAYG (0) or UnDP (1).}
2: $\mathcal{N}(population) \leftarrow$ initialize with randomly assigning 0 or 1 in B for each candidate participant partition in the population.
3: $i \leftarrow 0$
4: **while** $i < iter_{max}$ **do**
5: $\mathcal{K} \leftarrow keepbest(\mathcal{N})$ {The *best* partitions are the ones that can get the smallest values on *Refund* function.}
6: $\mathcal{C} \leftarrow crossover(\mathcal{N})$
7: $\mathcal{M} \leftarrow mutation(\mathcal{N})$
8: $\mathcal{N} \leftarrow \{\mathcal{K}, \mathcal{C}, \mathcal{M}\}$
9: $i \leftarrow i + 1$
10: **end while**
11: $U_p, U_u \leftarrow$ the *best* participant partition that can achieve the smallest *Refund* during all the iterations.

while $l_{i,j}$ is the number of the locations that participant i would visit during the sensing cycle j. In this paper, we predict $l_{i,j}$ via the mobility prediction method previously discussed.

In real life, the sensed data size distribution is likely to be more complicated than suggested by our estimation formula, due to our abstracting away from subtle variations between different instances of sensing tasks. In our future work, we will model the sensed data size in a more precise way, taking into account the different types of sensing tasks for different participants.

C. GA-Based Partition Algorithm

Based on the mobility prediction and sensed data size estimation results, we can approximate the organizer's 3G data refund budget for the upcoming month. This 3G data refund budget approximation function takes a specific UnDP/PAYG participant partition as input: $\text{Refund}(U_u, U_p)$.

In other words, given a participant partition result (i.e., knowing U_u and U_p), we can approximate Refund for the next upcoming month (supposing all the sensing cycles for the following month is C):

$$\text{Refund} = |U_u| * \text{Price}_u + \sum_{j \in C} \sum_{i \in U_p} d_{i,j} * \left(1 - p_{i,j}\right) * \text{Price}_p.$$

Considering Refund as the optimization function, participant partition can be directly expressed as a set split problem for minimizing Refund, which is *NP-hard. Thus, we use a genetic algorithm [31] to obtain a near-optimal solution. Algorithm 1 shows the pseudocode of the genetic algorithm. We use a bi-vector to mark each participant as a PAYG (0) or UnDP (1) participant (line 1). In each generation, we keep the best participant partitions from the previous generation by evaluating Refund function on every partition (line 5). In addition to the best partitions, applying crossover (line 6) and mutation (line 7) functions [31] on the previous generation's population, we obtain the other two sets of candidate partitions, in order to compose the whole population that will be examined in the new generation (line 8). Finally, the genetic algorithm could generate a near-optimal participant partition result. Note that we adopt a fixed number of iterations as the stopping criterion for the genetic algorithm (line 4).

VII. Evaluation

In this section, we first evaluate ecoSense using the MIT reality mining data set [7] including 48 active users. Then, to further evaluate both ecoSense's budget saving efficacy and algorithm computation efficiency with a larger number of users, we simulate a 500-user mobility trace leveraging SWIM [8] and briefly show the corresponding evaluation results.

A. Experimental Setup on MIT Reality Mining

For the MIT reality mining data set, we choose two months' data (October 2004 and November 2004) from 48 active users who have more than 20-day records per month. The mobility traces in the MIT reality data set include each user's Bluetooth encounters with other users and gateways deployed by the organizer. Note that to ensure that the sensed data can be successfully offloaded between two users via Bluetooth, we only use the Bluetooth encounters lasting for more than 5 min in the experiment. We use the first month's user data (October 2004) as the historic record to obtain a participant partition of PAYG and UnDP groups, and evaluate the performance on the second month (November 2004). The sensing cycle lasts for one day and the delayed-uploading cycle lasts for 3 h[6] (recall Fig. 1 for the illustration of sensing and delayed-uploading cycles). In this experiment, we set two Bluetooth gateways named localhost.media.mit.edu and studies.media.mit.edu. The costs of PAYG and UnDP price plans are set to \$0.1/MB and \$7/month, respectively.

1) Adding Up-to-Date WiFi Usage Logs: To mitigate the shortage of WiFi usage logs in the MIT data set,[7] we use an up-to-date mobile phone usage data set, Cambridge Device Analyzer [32], to complement the MIT data set. Specifically, for each MIT user u_{mit}, we randomly select a Cambridge user u_{cam} and uses u_{cam}'s latest two-month WiFi usage logs to represent u_{mit}'s WiFi usage.

2) Setting the Battery Threshold for Relaying Data: In our experiment, a user will relay others' data only when the battery level of her phone is above a predefined threshold (50%), because a user is typically not willing to relay data if her phone battery is low. Due to the lack of the battery information in the MIT data set, we simulate each user's phone battery level at any time based on the phone usage records, including calls, messages, and mobile data usages [33]. Interested readers can refer to our previous work [6, Sec. VI-C] for more details about this simulation.

B. Baseline Methods

To compare with ecoSense, we introduce the following two methods.

1) *Direct-Assignment:* Is the baseline method that assigns each participant to the UnDP or PAYG group directly according to her estimated sensed data size in the upcoming month.

[6]The delayed-uploading cycle starts at noon every day.

[7]Back to 2004 when the MIT reality mining campaign was conducted, WiFi was not a popular data transmission method for mobile phones.

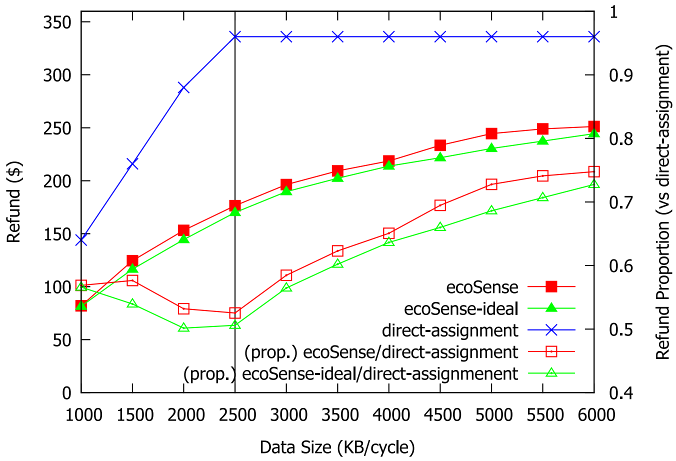


Fig. 5. Refund budget for fixed-size data uploading.

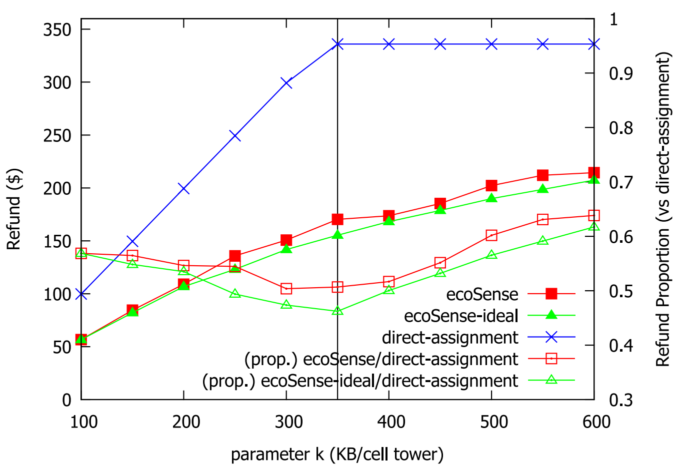


Fig. 6. Refund budget for varied-size data uploading ($c = 0$).

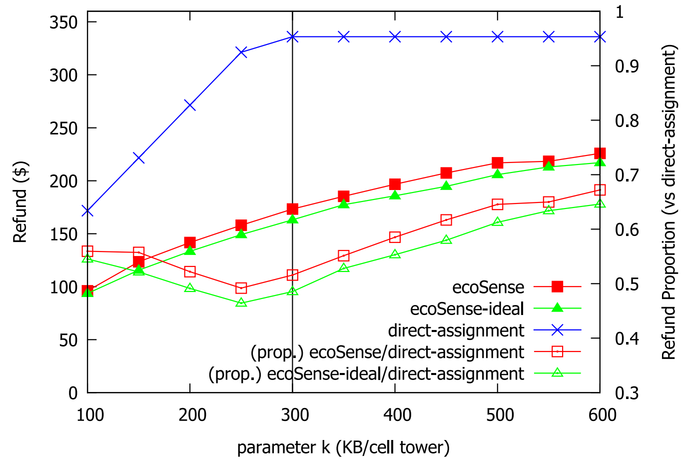


Fig. 7. Refund budget for varied-size data uploading ($c = 500$ KB).

2) *ecoSense-Ideal:* Does not do mobility prediction and directly leverages the second month's mobility trace (November 2004) in the genetic algorithm in order to get an optimal participant partition. In real life, these future data are not available; however, this result serves as a bound of the organizer's refund budget and shows the potential improvement that we could make by designing a better mobility prediction method in the future.

C. Evaluation Results on MIT Reality Mining

The most important issue is how much ecoSense could save for the organizer's 3G data refund budget when compared to direct-assignment. We will first show the results when all the participants upload the same size of sensed data in each sensing cycle (i.e., fixed-size data setting). Then, we will change the fixed-size data setting to the varied-size data setting, where the size of the sensed data that a participant would contribute is proportional to the number of visited locations. For the sake of simplicity, the above experiments are conducted using the OneRelay uploading strategy (the "uploading strategy" in the evaluation section refers to "PAYG uploading strategy"). Afterward, we will illustrate the difference between the OneRelay, OneHopFlooding, and Epidemic uploading strategies, from two perspectives: 1) refund budget and 2) energy efficiency. Finally, we evaluate how two parameters—max tolerable delay and 3G price plan cost—affect ecoSense's performance.

1) Fixed-Size Sensed Data Uploading: Here, we show how much refund cost ecoSense could save compared to direct-assignment for sensing tasks generating fixed-size sensed data. We adopt OneRelay as the uploading strategy. To see whether ecoSense's Poisson-distribution-based mobility prediction method has already achieved good enough performance or not, we also compare ecoSense with ecoSense-ideal.

Fig. 5 shows the mobile data cost refund under different fixed data size settings: 1000–6000 KB/cycle. ecoSense could save 25%–48% monetary refund compared to direct-assignment. We find that the data size setting where ecoSense can get the most significant effect (∼48% saving) is around 2500 KB/cycle, where direct-assignment changes from assigning the participants with PAYG to UnDP (denoted as turning point, the vertical line in Fig. 5).

Compared to ecoSense-ideal, ecoSense's refund budget is larger by only 1%–4%. This demonstrates that with the Poisson-distribution-based mobility prediction method, ecoSense has already achieved good performance. It may be possible to design a better mobility prediction method, but even with higher accuracy, the improvement on refund budget saving is not likely to be significant.

2) Varied-Size Sensed Data Uploading: We now consider the 3G data refund budget for varied-size sensing tasks. As mentioned in Section VI-B, the estimated varied data size for participant i in sensing cycle j can be modeled as $d_{i,j} = c + k * l_{i,j}$.

We write $l_{i,j}$ for the number of cell towers the participant i stayed in during sensing cycle j.[8] Fig. 8 illustrates the average number (avg ± std) of the visited cell towers per sensing cycle for each participant in November 2004. The average number of visited cell towers for different participants ranges from 3.6 to 9.7 per sensing cycle. For constant c in the $d_{i,j}$ model,

[8]To avoid those cell towers that a participant just passed by, we record a cell tower only if a participant stayed in the vicinity of the cell tower for more than 5 min.

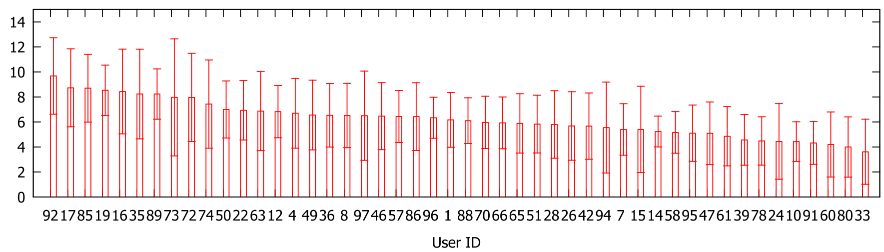


Fig. 8. Average number of visited cell towers per day for each participant in November 2004.

when $c = 0$, we assume that the participants sense only in each visited place. When $c \neq 0$, we assume that besides the sensed data for each visited place, participants also upload some other sensed data, e.g., aggregated coarse activity log, which accounts for approximately the same size of data for different participants (e.g., 500 KB).

Figs. 6 and 7 show the organizer's 3G data refund budget when the sensed data size for each cell tower is set as k ranges from 100 to 600 KB/cell tower, where c is set to 0 and 500 KB, respectively. Similar to the evaluation results for the fixed-size data setting, compared to direct-assignment, ecoSense could save 33%–51% of the refund budget. Besides, the most significant budget saving also appears around the turning point (the vertical lines in Figs. 6 and 7), where direct-assignment begins to assign all the participants to UnDP.

Compared to ecoSense-ideal, ecoSense's refund budget is larger by 1%–5%, which gives the range for improvement through the introduction of a more precise mobility prediction method.

3) Different Uploading Strategies: Previous experiments all run under the OneRelay uploading strategy. Here, we study how different uploading strategies (OneRelay, OneHopFlooding, or Epidemic)[9] would affect ecoSense's performance. In addition to the 3G data refund budget, we also consider the energy consumption in participants' smartphones by counting participants' relays and battery drain per delayed-uploading cycle. For simplicity, we consider only the fixed-size data setting.

Fig. 9 shows the 3G data refund budget when PAYG participants adopt different uploading strategies for fixed-size data. As expected, Epidemic achieves the smallest refund budget, while OneRelay incurs the largest. Specifically, Epidemic could save 2%–15% more budget than OneRelay. For smaller sensed data sizes, the improvement is more pronounced (e.g., 15% for 1000–1500 KB/cycle, while 2% for 6000 KB/cycle). This occurs because, for smaller data packets, fewer participants will be assigned to UnDP, leading to more PAYG participants. The large number of the PAYG participants consequently improve the performance of Epidemic. For OneHopFlooding, it achieves almost the same refund budget as Epidemic with the increase in data size. This indicates that with more UnDP participants for larger data packets, flooding a PAYG user's data within only one hop has already achieved high probability of making the data received by an UnDP participant or gateway (thus reducing the budget); using Epidemic, i.e., removing the restriction of one-hop of OneHopFlooding, does not increase this probability significantly.

[9]We set the reconnection time threshold of Epidemic to one hour.

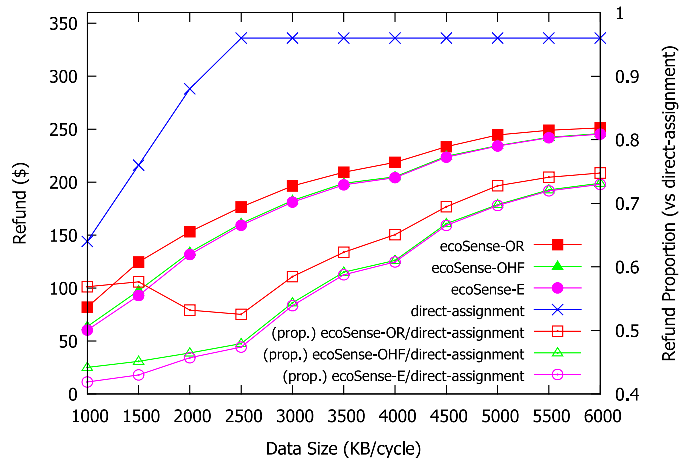


Fig. 9. Refund budget for different uploading strategies (OR: OneRelay, OHF: OneHopFlooding, and E: Epidemic).

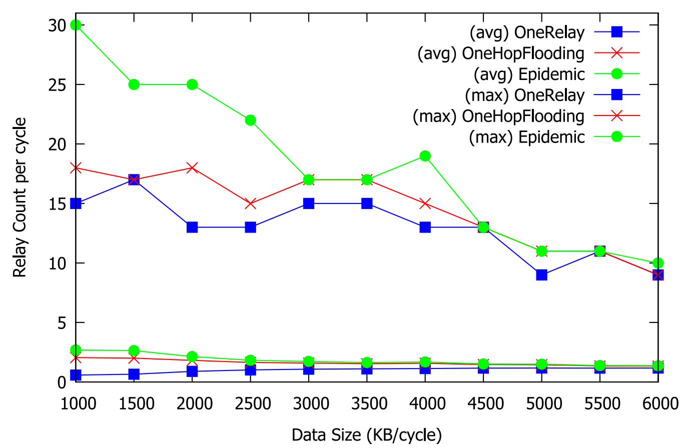


Fig. 10. Relay count for different uploading strategies.

Fig. 10 shows the average relay count per delayed-uploading cycle for a participant, as well as the maximum relay count among all the participants (including both PAYG and UnDP groups). Epidemic and OneHopFlooding incur the similar average relay count, which is 1.2–4.6 times larger than OneRelay. Considering the maximum relay count for a participant in a delayed-uploading cycle (i.e., the worst energy consumption case), Epidemic is significantly larger than OneHopFlooding and OneRelay, especially when the data

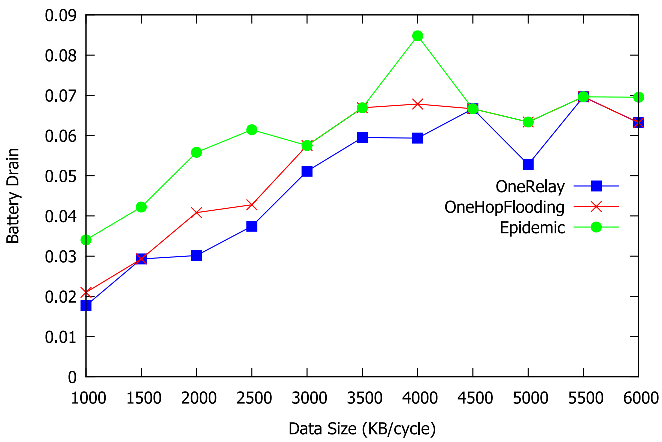


Fig. 11. 2000 mAh battery drain of the worst participant for different uploading strategies.

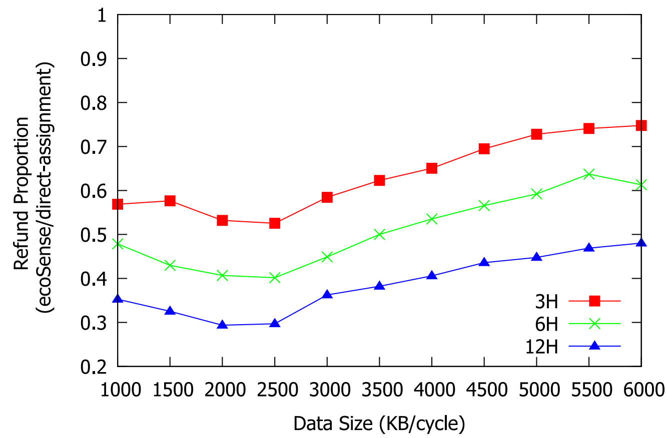


Fig. 12. Refund budget proportions for varying maximum tolerable delays.

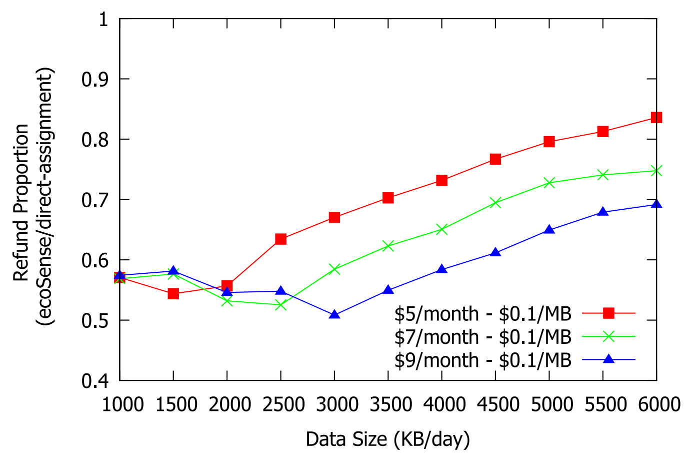


Fig. 13. Refund budget proportions for varying price settings.

size is less than 3000 KB/cycle. For example, when data size is 1000 KB/cycle, the maximum relay count for Epidemic is 30, while only 18 and 15 for OneHopFlooding and OneRelay, respectively. The maximum relay count for OneHopFlooding is also larger than OneRelay; however, the difference is usually smaller than the average relay count (less than 1.5 times). Specifically, using OneHopFlooding would trigger at most five more relays than OneRelay for the "worst participant" who consumes the most energy for relays. Based on previous research on mobile phone energy consumption [33], we calculate the mobile phones' battery drain for the worst participant with different uploading strategies. Fig. 11 shows that—for a 2000 mAh battery—the battery drain with OneHopFlooding and OneRelay is always less than 7%, while the battery drain difference between the two strategies is at most 2%; in comparison, the battery drain with Epidemic is a bit higher, leading to 8.5% in the worst case. For real-life scenarios, the strategy selection among Epidemic, OneHopFlooding, and OneRelay needs to be further studied in order to better balance the 3G refund saving and the participant mobile phones' battery drain.

4) Other Experimental Parameter Analysis: To better understand how max tolerable delay and 3G price plan cost affect ecoSense's performance, in this section, we use some different parameter settings from the ones used in the previous experiments.

a) Varying maximum tolerable delays: Here, we conduct the experiments to test ecoSense's performance for various delays other than 3 h. Suppose the uploading strategy is OneRelay, Fig. 12 shows the 3G data refund budget proportion (ecoSense versus direct-assignment) for 3/6/12 h maximum delays. As the delay increases, the 3G refund budget becomes lower. This is because adopting a longer delay, participants have more opportunities to relay data via another participant or a Bluetooth/WiFi gateway to reduce 3G data cost.

b) Varying 3G price plan costs: The costs of different telecom operators' 3G price plans can vary. Thus, we also examine the evaluation results for various price plan settings.

Suppose the uploading strategy is OneRelay, Fig. 13 shows the refund budget proportion (ecoSense versus direct-assignment) for three different price plan settings—where PAYG prices are all set to \$0.1/MB and UnDP prices are set to \$5, \$7, and \$9/month, respectively. The most significant difference among different settings is that the turning point (i.e., the data size where the refund budget proportion is the lowest) changes, because the turning point is the sensed data size where direct-assignment begins to assign all the participants to UnDP, which is sensitive to the price plan setting. Except for the turning point, most observations are similar for different settings. For example, whatever the price plan settings, ecoSense could save at most about 50% of the refund budget compared to direct-assignment.

D. Experiment on SWIM Simulation

To evaluate ecoSense's budget saving efficacy and algorithm computation efficiency (especially the participant partition algorithm) on a larger number of users, we simulate a user-encounter data set containing 500 users' 2-month traces by SWIM [8]. The simulation parameters are selected according to the "Dartmouth" setting in [8], in order to simulate user encounters in a campus. The experimental settings are similar to those used with the MIT reality mining data set: the sensing cycle lasts for one day and delayed-uploading cycle lasts for 3 h. For simplicity, we show only the evaluation results when the uploading strategy is OneRelay and the sensing task is fixed-size.

TABLE I
REFUND BUDGET FOR 500-USER SWIM SIMULATION

data size	*ecoSense*	*direct-assignment*	*saving ratio*
500 KB	\$368	\$750	51%
1500 KB	\$573	\$2250	75%
2500 KB	\$728	\$3500	79%
3500 KB	\$849	\$3500	76%
4500 KB	\$956	\$3500	73%

1) Budget Saving Efficacy: Table I shows the refund budget with the 500-user simulation data set. As the user number increases to 500 and the users' activity area remains within the campus-like scale of the MIT data set, the user spatial density in the SWIM simulation is much larger than that for the MIT data set, which leads to more refund budget saving. For example, when the fixed data size is 2500 KB, ecoSense can save 79% in the 500-user simulation data set, with respect to 48% on the 48-user MIT data set (see Fig. 5). In other words, if more users are active in a smaller area, they will have more chances to meet each other so that ecoSense could save more 3G data cost.

2) Algorithm Computation Efficiency: With respect to computation efficiency, we focus on the run-time performance of the server-side participant partition component, as the smartphone-side uploading decision making component's strategy (OneRelay, OneHopFlooding, or Epidemic) is simple. For the participant partition component, the genetic algorithm dominates the running time, because our proposed mobility prediction and sensed data size estimation methods are simple and run much faster than the genetic algorithm.

For the genetic algorithm, the main difference between 48 users (MIT) and 500 users (SWIM simulation) is that for more users we need more iterations to obtain a near-optimal result. In our experiment environment,[10] we can get a good solution in 50 rounds for the 48-user MIT data set (approximately 5 min of execution time), while we need 500 rounds for the 500-user simulation data set (approximately 90 min of execution time). As the participant partition is an offline algorithm, which only needs to run once a month, we believe that this execution efficiency is already adequate for most real-life conditions. Furthermore, the performance of the genetic algorithm can be easily (and dramatically) improved by leveraging its inherent parallelism.

VIII. DISCUSSION

As this paper is the first research investigating how to refund crowdsensing participants' 3G data usage incurred by sensing tasks and it is still at an early age, we will discuss some issues which are not addressed due to space limit, and point out some future potential research directions.

A. Energy Consumption Issues

In addition to the 3G data cost discussed in this paper, energy consumption is another critical concern for MCS participants. It is worth noting that with the popularity of Bluetooth 4.0 low energy technology, the energy consumption of the Bluetooth scanning becomes much lower [34], which makes ecoSense more applicable nowadays. Actually, some novel applications requiring the Bluetooth scanning always on have already been off-the-shelf, e.g., real-time fitness sensing with FitBit wristbands.[11]

Besides, some energy-efficient mechanisms can still be incorporated into ecoSense, such as piggybacking [35] and uploading data when the phone is charging. In our future work, we will study the possibility and feasibility of introducing these energy-efficient mechanisms into the ecoSense framework, in order to improve both PAYG and UnDP participants' experience.

B. Other Kinds of Monetary Incentives

In this paper, we consider only the refund to cover crowdsensing participants' additional 3G data cost. In real life, other kinds of monetary incentives can also be provided to the participants, such as a fixed payment for the participation and higher reward (prize) for a small number of most active participants.

Though the refund of 3G data cost is only a part of the total monetary incentives, it could be a reasonable baseline because refunding each participant with her 3G data cost can mitigate the participant's worry about whether the 3G data usage would exceed her data plan and incur extra fees [3], [4]. In other words, refunding 3G data cost could at least prevent the participants from paying extra fees because of participation in a crowdsensing task. Other kinds of monetary incentives could be added to the 3G data cost refund to further encourage users to participate in the sensing task.

C. Participants' Personal 3G Data Usage

In addition to the 3G data usage for the crowdsensing task, participants also consume 3G data for their personal application usage. Here, we prove that ecoSense can always work effectively (i.e., refund can cover additional 3G cost incurred by the crowdsensing task) even if we do not know participants' actual personal data usage.

For an UnDP participant, personal data usage does not matter because the unlimited 3G data plan covers it automatically. However, for a PAYG participant, each of the two price plan cases requires further clarification and analysis.

1) If a PAYG participant u_p's original price plan is also PAYG (i.e., small personal data usage), then ecoSense's refund for u_p just equals the additional 3G data cost incurred by the sensing task.
2) If a PAYG participant u_p's original price plan is UnDP (i.e., large personal data usage), obviously u_p's reasonable choice is still using UnDP as price plan after participating in the sensing task, which means that u_p's additional 3G data cost incurred by the sensing task is 0. Currently, ecoSense would still refund this kind of participant through the PAYG scheme (i.e., refunding them the money proportional to their 3G-uploaded

[10] Software: DEAP (https://github.com/DEAP/deap) with python 2.7, Windows 7; Hardware: Intel core i7-3612QM@2.1 GHz, 8G RAM.

[11] https://www.fitbit.com/

data size), because we cannot easily distinguish them from the previous small-personal-data-usage kind of participant.[12]

In summary, without knowing the actual personal data usage for each participant, ecoSense's refund mechanism can always cover each participant's additional 3G data cost incurred by the crowdsensing task.

D. Other 3G Price Plans

Usually, telecom operators offer users 3G price plans other than PAYG and UnDP. For example, D100 MB price plan: $1 for the first 100 MB data and then $0.1/MB (like PAYG). Here, we discuss the problem: whether ecoSense's refund mechanism can still cover participants' additional 3G cost when other 3G price plans exist.

On the one hand, by still using only two refund schemes of PAYG and UnDP, even if other 3G price plans exist, ecoSense can effectively refund participants to cover their additional 3G cost. For example, assume that a participant's original price plan is D100MB. If refunded as UnDP, she can change next month's price plan to UnDP; and if refunded as PAYG, she can keep the D100MB price plan. As long as a participant follows the above rule, ecoSense's refund can always cover her additional 3G cost.

On the other hand, it is promising to introduce new 3G price plans, such as D100MB, into refund schemes of ecoSense. For example, after importing D100MB as a new refund scheme (i.e., refund schemes increase to three types: UnDP, D100MB, and PAYG), we could model the participants with D100MB refund scheme as "UnDP" participants when data usage is less than 100 MB, and as "PAYG" participants when data usage exceeds 100 MB. This type of polymorphic participant partition mechanism requires further research.

IX. Conclusion

Refunding MCS participants for additional 3G data cost incurred during the crowdsensing process is an effective marketing strategy for the MCS organizer. In this paper, we investigate the problem of how to minimize the total 3G data refund budget for the crowdsensing organizer who follows such a marketing strategy. Based on two widely used 3G price plans, i.e., PAYG and UnDP, we propose a delay-tolerant data uploading framework called ecoSense, whose goal is to minimize the organizer's 3G refund budget for all the participants.

By introducing delay-tolerant data uploading mechanisms, UnDP participants could relay PAYG participants' sensed data to the server without additional 3G cost; PAYG participants could also upload their sensed data via free-charge Bluetooth/WiFi gateways to reduce 3G cost. Based on these observations, we propose the data uploading strategies for both PAYG and UnDP participants and design a participant partition algorithm to determine whether a participant should be assigned to PAYG or UnDP. Our ecoSense framework was evaluated using the MIT Reality Mining data set and a larger SWIM simulation data set. The evaluation results showed that ecoSense could save up to $\sim$50% of the refund budget compared to direct-assignment that assigns each participant to UnDP or PAYG directly according to the size of her sensed data.

[12]We can distinguish these two kinds of participants provided we assume that all the participants trustfully report their original price plans. However, due to the privacy concern and user selfishness, we do not make this assumption in our current work.

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.

[2] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4W1H in mobile crowd sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 42–48, Aug. 2014.

[3] T. Luo and C.-K. Tham, "Fairness and social welfare in incentivizing participatory sensing," in *Proc. SECON*, Seoul, Korea, 2012, pp. 425–433.

[4] E. Massung, D. Coyle, K. F. Cater, M. Jay, and C. Preist, "Using crowdsourcing to support pro-environmental community activism," in *Proc. CHI*, Paris, France, 2013, pp. 371–380.

[5] L. Wang, D. Zhang, and H. Xiong, "effSense: Energy-efficient and cost-effective data uploading in mobile crowdsensing," in *Proc. UbiComp Adjunct*, Zürich, Switzerland, 2013, pp. 1075–1086.

[6] L. Wang, D. Zhang, Z. Yan, H. Xiong, and B. Xie, "effSense: A novel mobile crowd-sensing framework for energy-efficient and cost-effective data uploading," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 12, pp. 1549–1563, Dec. 2015.

[7] N. Eagle and A. Pentland, "Reality mining: Sensing complex social systems," *Pers. Ubiquitous Comput.*, vol. 10, no. 4, pp. 255–268, 2006.

[8] S. Kosta, A. Mei, and J. Stefa, "Large-scale synthetic social mobile networks with SWIM," *IEEE Trans. Mobile Comput.*, vol. 13, no. 1, pp. 116–129, Jan. 2014.

[9] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: An end-to-end participatory urban noise mapping system," in *Proc. IPSN*, Stockholm, Sweden, 2010, pp. 105–116.

[10] Y. Chon, E. Talipov, and H. Cha, "Autonomous management of everyday places for a personalized location provider," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 518–531, Jul. 2012.

[11] K. K. Rachuri, C. Mascolo, M. Musolesi, and P. J. Rentfrow, "SociableSense: Exploring the trade-offs of adaptive sampling and computation offloading for social sensing," in *Proc. MobiCom*, Las Vegas, NV, USA, 2011, pp. 73–84.

[12] G. Cardone, A. Cirri, A. Corradi, and L. Foschini, "The ParticiPact mobile crowd sensing living lab: The testbed for smart cities," *IEEE Commun. Mag.*, vol. 52, no. 10, pp. 78–85, Oct. 2014.

[13] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. MobiCom*, Istanbul, Turkey, 2012, pp. 173–184.

[14] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment framework for participatory sensing data collections," in *Pervasive Computing*. Heidelberg, Germany: Springer, 2010, pp. 138–155.

[15] D. Zhang, H. Xiong, L. Wang, and G. Chen, "CrowdRecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *Proc. UbiComp*, Seattle, WA, USA, 2014, pp. 703–714.

[16] H. Xiong, D. Zhang, G. Chen, L. Wang, and V. Gauthier, "CrowdTasker: Maximizing coverage quality in piggyback crowdsensing under budget constraint," in *Proc. PerCom*, St. Louis, MO, USA, 2015, pp. 55–62.

[17] H. Xiong *et al.*, "iCrowd: Near-optimal task allocation for piggy-back crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 2010–2022, Aug. 2016.

[18] L. Wang *et al.*, "CCS-TA: Quality-guaranteed online task allocation in compressive crowdsensing," in *Proc. UbiComp*, Osaka, Japan, 2015, pp. 683–694.

[19] A. Albers, I. Krontiris, N. Sonehara, and I. Echizen, "Coupons as monetary incentives in participatory sensing," in *Collaborative, Trusted and Privacy-Aware e/m-Services*. Heidelberg, Germany: Springer, 2013, pp. 226–237.

[20] W. Sherchan *et al.*, "Using on-the-move mining for mobile crowdsensing," in *Proc. MDM*, Bengaluru, India, 2012, pp. 115–124.

[21] M. Musolesi, M. Piraccini, K. Fodor, A. Corradi, and A. T. Campbell, "Supporting energy-efficient uploading strategies for continuous sensing applications on mobile phones," in *Pervasive Computing*. Heidelberg, Germany: Springer, 2010, pp. 355–372.

[22] B. Hull *et al.*, "CarTel: A distributed mobile sensor computing system," in *Proc. SenSys*, Boulder, CO, USA, 2006, pp. 125–138.

[23] H. Liu *et al.*, "Efficient 3G budget utilization in mobile participatory sensing applications," in *Proc. INFOCOM*, Turin, Italy, 2013, pp. 1411–1419.

[24] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell, "NextPlace: A spatio-temporal prediction framework for pervasive systems," in *Pervasive Computing*. Heidelberg, Germany: Springer, 2011, pp. 152–169.

[25] Y. Chon, H. Shin, E. Talipov, and H. Cha, "Evaluating mobility models for temporal prediction with high-granularity mobility data," in *Proc. PerCom*, Lugano, Switzerland, 2012, pp. 206–212.

[26] R. I. Ciobanu and C. Dobre, "Predicting encounters in opportunistic networks," in *Proc. HP-MOSys*, Paphos, Cyprus, 2012, pp. 9–14.

[27] H. Xiong, D. Zhang, L. Wang, and H. Chaouchi, "EMC3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint," *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1355–1368, Jul. 2015.

[28] H. Xiong, D. Zhang, L. Wang, J. P. Gibson, and J. Zhu, "EEMC: Enabling energy-efficient mobile crowdsensing with anonymous participants," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, 2015, Art. ID 39.

[29] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. WDTN*, 2005, pp. 252–259.

[30] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Dept. Comput. Sci., Duke Univ., Tech. Rep. CS-200006, 2000.

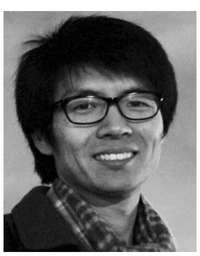
[31] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[32] D. T. Wagner, A. Rice, and A. R. Beresford, "Device analyzer: Understanding smartphone usage," in *Proc. MobiQuitous*, Tokyo, Japan, 2014, pp. 195–208.

[33] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: A measurement study and implications for network applications," in *Proc. IMC*, Chicago, IL, USA, 2009, pp. 280–293.

[34] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of Bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11734–11753, 2012.

[35] N. D. Lane *et al.*, "Piggyback crowdsensing (PCS): Energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities," in *Proc. SenSys*, Rome, Italy, 2013, pp. 1–14.



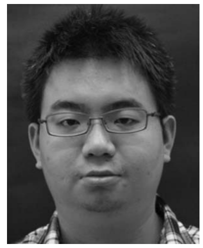
Leye Wang received the B.Sc. and M.Sc. degrees in computer science from Peking University, Beijing, China. He is currently pursuing the Ph.D. degree with the Institut Mines-Télécom/Télécom SudParis, Evry, France, and the Université Pierre et Marie Curie, Paris, France.

His research interests include mobile crowdsensing and ubiquitous computing.

Daqing Zhang (M'11) received the Ph.D. degree from the University of Rome "La Sapienza," Rome, Italy, and the University of L'Aquila, L'Aquila, Italy, in 1996.

He is a Professor with Télécom SudParis, Institut Mines-Télécom, Evry, France. He has published over 200 technical papers in leading conferences and journals, where his work on context model is widely accepted by the pervasive computing, mobile computing, and service-oriented computing communities. His current research interests include context-aware computing, urban computing, mobile computing, big data analytics, and pervasive elderly care.

Dr. Zhang was a recipient of the Ten-Years CoMoRea Impact Paper Award at the IEEE PerCom 2013, the Honorable Mention Award at the ACM UbiComp 2015, the Best Paper Award at the IEEE UIC 2015 and 2012, and the Best Paper Runner Up Award at Mobiquitous 2011. He served as the General or Program Chair for over 10 international conferences, giving keynote talks at over 16 international conferences. He is an Associate Editor of the *ACM Transactions on Intelligent Systems and Technology* and the IEEE TRANSACTIONS ON BIG DATA.



Haoyi Xiong (M'15) received the B.Eng. degree in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, the M.Sc. degree in information technology from the Hong Kong University of Science and Technology, Hong Kong, and the Ph.D. degree from Pierre and Marie Curie University, Paris, France, and the Institut Mines-Télécom/Télécom SudParis, Evry, France, in 2015.

He is currently a Research Associate with the Department of Systems and Information Engineering, University of Virginia, Charlottesville, VA, USA. His research interests include mobile crowdsensing, human mobility modeling, and analysis.

Dr. Xiong was a recipient of the Best Paper Award from the IEEE UIC 2012. He is currently a member of the IEEE Computer Society and Communication Society.

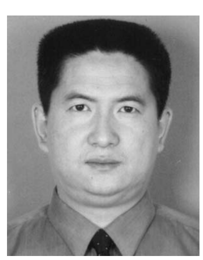
John Paul Gibson received the Ph.D. degree from Stirling University, Stirling, U.K., in 1993 and the Habilitation à Diriger des Recherches degree from the Université Henri Poincaré, Nancy, France, in 2011. The thesis demonstrated the need for a strong link between theory, practice, and teaching in the domain of formal software engineering.

He is currently with Telecom SudParis, Evry, France, which is part of the group of Grand Ecoles "Mines-Télécom." His research has examined a wide range of issues, including formal methods, object oriented modeling, software product lines, service engineering, e-voting, models of computation, innovative teaching techniques, and computer science education in schools.

Chao Chen received the B.Sc. and M.Sc. degrees in control science and control engineering from Northwestern Polytechnical University, Xi'an, China, in 2007 and 2010, respectively, and the Ph.D. degree from Pierre and Marie Curie University, Paris, France, and the Institut Mines-Télécom/Télécom SudParis, Evry, France, in 2014.

He is an Associate Professor with the College of Computer Science, Chongqing University, Chongqing, China. In 2009, he was a Research Assistant with Hong Kong Polytechnic University, Hong Kong. His research interests include pervasive computing, social network analysis, data mining from large-scale taxi data, and big data analytics for smart cities.

Dr. Chen was a co-recipient of the Best Paper Runner-Up Award at MobiQuitous 2011.



Bing Xie was born in Hunan, China, in 1970. He received the Ph.D. degree in computer science from the Changsha Institute of Technology, Hunan, in 1998.

He was a Post-Doctoral Fellow with Peking University, Beijing, China, in 1998, where he was an Associate Professor until 2006. He is currently a Professor with the Department of Computer Science, Peking University. He has authored or coauthored over 60 international publications. He has hosted nine research projects, as PI, supported by the Chinese government concerning software engineering and formal method. His research interests include software reuse, ubiquitous computing, and its industrial application.