Information-Centric Collaborative Data Collection for Mobile Devices in Wireless Sensor Networks

Gang Xu[1], Edith C.-H. Ngai[1] and Jiangchuan Liu[2]
[1] Department of Information Technology, Uppsala University, Sweden
[2] School of Computing Science, Simon Fraser University, Canada

***Abstract*—The advancement of smart phones enables mobile users to collect data from their surrounding sensors using short-range wireless communication. However, the limited contact time and the wireless capacity constrain the amount of data to be collected by the mobile users. It is crucial for mobile users to collect sensing data that can maximize their data utility. In this paper, we propose a distributed algorithm to provide information-centric ubiquitous data collection for multiple mobile users. The mobile users construct data collection trees adaptively according to their dynamic moving speeds. They prioritize data collection according to the information value carried by the sensing data. The distributed algorithm can support smooth data collection and coordination among multiple mobile users. We evaluate the data utility, energy efficiency and scalability of our solution with extensive simulations. The results showed that our distributed algorithm can improve information value up to 50% and reduce energy consumption to half compared with the existing approach.**

I. Introduction

Wireless sensors networks (WSNs) have been widely deployed for environmental monitoring to build a sustainable society. They enable better understanding of climate change, pollution, and habitats from natural environment to water system and transportation in urban life [1], [2], [3]. With the advancement of mobile devices, ubiquitous data collection allows mobile users to collect data from their nearby sensors using their mobile devices, such as PDAs or smart phones [4], [5], [6]. This architecture increases the flexibility of sensor network deployment and offers a cost-effective solution for sensor data collection. Mobile users can support data collection in sensing applications such as environmental monitoring, social network, healthcare, transportation, and safety, etc. For example, PEIR (Personal Environmental Impact Report) is a mobile sensing application developed to calculate personalized estimates of environmental impact and exposure [7]. Since the available sensor types are limited on mobile phones, mobile users can collect sensing data with more variety by communicating with wireless sensors in their surroundings.

Different from traditional WSNs, ubiquitous data collection does not rely on a stationary sink to collect sensing data from the whole network. Instead, mobile users collect sensing data from their surrounding sensors pervasively using their mobile devices. The uncontrolled mobility of the mobile users and the limited wireless communication range pose new challenges in ubiquitous sensor data collection. In particular, the contact time between the mobiles and the sensors can be short given the continuous and potentially fast movement of the mobile users. It is crucial for the mobile users to maximize their information gain under the limitations of the contact time and the wireless capacity.

Although data collection for WSNs with mobile elements has been studied, the previous work tackled the problem mostly in a controllable environment. Most studies considered mobile sinks moving along predefined paths and collecting data at rendezvous points. Different from mobile sinks, mobile users have uncontrolled and random mobility patterns. The data collection trees need to be constructed and migrated quickly to adapt to the mobility of the mobile users. Global data collection trees have been adopted by the mobile elements (MEs) in existing work [4], [8], [9], [10]. The idea is to broadcast HELLO messages to the nodes in the network for tree construction. However, it is difficult to determine how far the messages should be broadcast. A simple way to limit the size of the data collection tree is by predefining a maximum hopcount h in broadcast. Nevertheless, this approach does not consider the dynamic moving speed of the mobile user. In addition, the mobile users may not have enough contact time to collect all the data from the surrounding sensors. Hence, it is important to collect data that contain the most valuable information for the users. In this work, we propose a distributed approach to schedule data collection to maximize their information gain from the collected data.

We highlight the contributions of this work as follows. First, we propose a distributed algorithm, called EQRoute, to provide information-centric ubiquitous data collection with mobile users. It supports data collection for multiple mobile users with uncontrolled mobility in a distributed manner. Second, the mobile users can estimate their available capacity for data collection dynamically according to their moving speeds. Our algorithm provides energy-efficient and smooth data collection that maximizes the information gain for the users with low energy consumption. Finally, we evaluate the performance of EQRoute by extensive simulations. Compared with the most advanced existing approach, EQRoute collects data with much higher information values and reduces the energy consumption significantly.

The rest of this paper is organized as follows. The related work is presented in Section II. We describe the challenges and design goals of information-centric ubiquitous data collection in Section III. In Section IV, we present the problem formulation and a centralized optimal approach. We propose our distributed ubiquitous data collection algorithm in Section V. We evaluate the performance of our solution with extensive

978-1-4799-2003-7/14/$31.00 ©2014 IEEE

simulations in Sections VI. We conclude this paper in Section VII.

II. Related Work

Mobile sinks and mobile relays have been suggested for improving the performance of data collection in wireless sensor networks. Shah et al. [9] presented an architecture using moving entities, called *data mules*, to collect sensing data. Gu et al. [11] proposed a partitioning-based algorithm to schedule the movement of mobile elements, which minimizes the required moving speed and eliminates buffer overflow. Bisnik et al. [12] studied the problem of providing quality coverage using mobile sensors and analyzed the effect of controlled mobility on the fraction of events captured. Xu et al. [13] further studied delay tolerant event collection in sensor networks with mobile sink which considers the spatial-temporal correlation of events in the sensing field. He et al. [14] analyzed the performance of data collection theoretically to evaluate service disciplines of mobile elements through a queueing model. Nevertheless, the above works focus on controlling the movement of mobile sinks for data collection, which are different from the mobile users with independent and uncontrolled mobility in our work.

Apart from that, some studies have been conducted for mobile elements without any fixed trajectory. Kusy et al. [15] presented an algorithm to predict the mobility pattern of the mobile sinks from the training data. They computed and maintained the mobility graph of the mobile sinks to improve routing reliability in data collection. Similarly, Lee et al. [16] presented a routing scheme that exploits the mobility pattern of the mobile sinks to minimize energy consumption and network congestion. However, the above works emphasize on predicting the movement of mobile elements to improve routing efficiency. Recently, ubiquitous data collection with mobile users has been studied for mobile users to collect data from wireless sensors networks. Li et al. [4] proposed a ubiquitous data collection scheme that can efficiently form a new data collection tree by locally modifying the previously constructed data collection tree. However, the above work has been focusing on the reliability and energy efficiency on data collection. The information value carried by the sensing data has not yet been fully considered in data collection for mobile sensor networks.

III. Preliminary

A. Application Scenarios

We first consider a sensing field shown in Figure 1, where a mobile user is walking in a sensing field to collect data, for example, temperature and pollution. The wireless sensors take sensor measurements and store the data in their buffers. The data can be picked up by the mobile users when they pass by the sensors. We use *information value* to indicate the importance of various observations carried by the sensing data [17]. It is a normalized value between 0 and 1, i.e. $w = [0, 1]$. The information value is higher if the data carry more important information, such as abnormal temperature

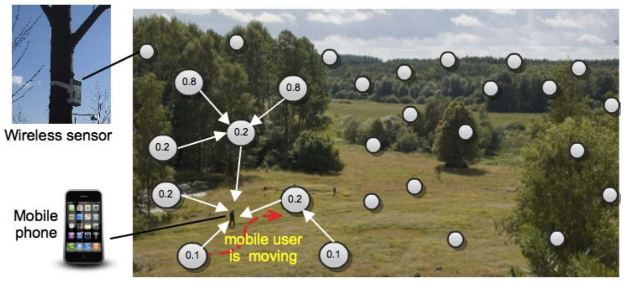


Fig. 1: A mobile user is walking in a sensing field to collect data from his surrounding sensors. The numbers in the figure indicate the information value, in the range of [0,1], carried by the sensing data. The data with higher information value are prioritized for data collection to maximize the information gain.

or high pollution level. Intuitively, the mobile users want to collect data that can maximize their information values. For example, in Figure 1, the mobile user will collect data with $w = 0.8$ rather than data with $w = 0.1$ if he has only limited capacity for data collection.

B. Challenges and Design Goals

Communication opportunity occurs between the mobile phones and the wireless sensors only when they are within the communication range of each other. Hence, it is challenging to collect data from the wireless sensors when the mobile user is moving with uncontrolled and unpredictable mobility. Due to the limited wireless communication range, i.e. IEEE 802.15.4 or bluetooth, the mobile user may have *short contact time* to collect sensing data especially when he is moving fast and without any stops. Since the sensors and the mobile device communicate over the same wireless channel, they have to share the *limited wireless capacity* with their neighbouring nodes. Bottleneck may occur particularly at the mobile node, since it is the root of the data collection tree that receives and processes maximum amount of traffic. Due to the above reasons, it is crucial for the users to collect data that can maximize the total information value. Distributed and localized approach is preferred to reduce communication overhead for coordination among multiple mobile users. We highlight the design goals of our distributed algorithm here:

- It provides information-centric ubiquitous data collection to maximize the information value from the collected data with low energy consumption.
- It supports smooth data collection adaptive to the moving speeds of the mobile users.
- It coordinates data collection with multiple mobile users in a distributed manner.

IV. Information-Centric Ubiquitous Data Collection

In this problem, the sensor nodes generate sensing data periodically and cache them in their buffers. The mobile devices with compatible wireless components, i.e. IEEE 802.15.4 or bluetooth, can collect data from their surrounding sensors. The collected data can be uploaded by the mobile devices to the server when the Internet connection is available later.

We focus on data collection from the wireless sensors to the mobile users in this study. Our goal is to maximize the information value and reduce the communication overhead in ubiquitous data collection. The mobile user is modelled as a mobile element (ME) with uncontrolled mobility, limited communication range, and variable moving speed. Each sensor can communicate with the MEs if they are within its wireless communication range. Multihop routing is supported to deliver data from the wireless sensors to the mobile users.

A. Problem Formulation

We introduce the following notations in our problem formulation:

- Each mobile user j creates a routing tree T_j.
- Each data packet d_i has an information value w_i.
- We use w_H and w_L to represent the information value of high and low priority data respectively.
- We use p_H and p_L to represent the packet generation probability of high and low priority data respectively.
- We denote μ_j as the maximum service rate that j can receive and process data from T_j.
- We define capacity C_j as the amount of data (in number of packets) that can be collected by j in a timeslot t, where $C_j = \mu_j t$. Note that this capacity is shared among the neighbouring nodes of j and their subtrees.
- We use hopcount c_{ij} to measure the communication cost for delivering d_i to j.

Variable:
$x_{ij} \in [0, 1]$: indicates whether packet d_i is sent using tree T_j.

Objective:

$$\max\ U = \sum_{i,j} \frac{w_i}{c_{ij}} x_{ij} \tag{1}$$

Constraints:

$$\sum_j x_{ij} \leq 1,\ \forall i \tag{2}$$

$$0 \leq x_{ij} \leq 1,\ \forall i, j \tag{3}$$

$$\sum_i x_{ij} \leq C_j,\ \forall j \tag{4}$$

We measure the information gain per communication cost of a data packet d_i by $u_i = w_i/c_{ij}$, which is the information value of the data divided by its communication cost in hopcount. Our objective function is to maximize the sum of u_i from all the collected data, denoted by U. This allows us to maximize the information value from the collected data, while achieving a good balance between the information value and the communication cost. Constraint 2 ensures that each data packet d_i is sent to only one mobile user. Constraint 3 allows fractional data to be sent in a packet. Constraint 4 ensures that the total packets received by mobile user j does not exceed its capacity C_j in a given time slot.

B. Optimal Centralized Algorithm

We suggest a centralized data collection algorithm for mobile user to maximize U from the collected data (see Algorithm 1). The mobile user first floods the network to gather the information value, size and communication cost of the data from the sensors. Then, it assigns capacity to the sensors by selecting data with the maximum information value per communication cost, i.e. w_i/c_{ij}. However, this algorithm works only in a centralized manner. The mobile user has to wait for the information from all the sensors before allocating the capacity to individuals. As discussed before, the mobile user has to flood the whole networks or broadcast to h hops, but h is not easy to decide.

Algorithm 1 Centralized Capacity Allocation

C_j: capacity of mobile j;
w_i: information value of data d_i;
c_{ij}: hopcount for d_i to reach mobile j;

Mobile user j broadcasts to all sensors in the network;
Each sensor replies to j with w_i and c_{ij} of its data;
while $C_j > 0$ **do**
  Choose the data d_i with maximum w_i/c_{ij};
  if $C_j \geq 1$ **then**
    $x_{ij} = 1$;
  else
    $x_{ij} = C_j$;
  end if
  $C_j = C_j - x_{ij}$;
end while

Theorem 1. *The above greedy algorithm gives an optimal solution for capacity allocation to sensors.*

Proof. The algorithm allocates the capacity C_j completely for receiving the sensing data. We assumed that the capacity is limited, so that it is not enough to collect all the data in the network. This implies that here exists a q, such that $1 = x_1 = ... = x_{q-1} > x_q \geq x_{q+1} = ... = 0$, where $x_{n+1} = 0$. We show the optimality of this solution by comparing to any other feasible solution $y_1, ..., y_n$ of this problem. Since w_i/c_{ij} are positive for all i, this solution can only be optimal if $\sum_i y_i = C_j$. Let k be the smallest index such that $y_k < 1$, and let l be the smallest index with $k < l$ such that $y_l > 0$. Note that such an l exists, unless the solution $y_1, ..., y_n$ is equal to the solution $x_1, ..., x_n$ obtained by the above greedy algorithm. We will now increase y_k and decrease y_l, while keeping all other values equal, to obtain a new solution. Let $\epsilon = \min\{1 - y_k, y_l\} > 0$. Increase y_k by ϵ and decrease y_l by ϵ. It is easy to find that this move yields a feasible solution with value not smaller than the value of the solution $y_1, ..., y_n$. Moreover, either y_k has become equal to 1, or y_l has become equal to 0. Repetition of this argument eventually yields the solution $x_1, ..., x_n$ obtained by the greedy algorithm. $\square$

However, the energy consumption is very high in this centralized approach. The mobile user has to broadcast to all the nodes in the network and get back their replies. The communication overhead is in the order of $O(N)$, where N is the number of nodes in the network.

V. Distributed Algorithm Design

We propose a distributed and information-centric ubiquitous data collection algorithm, called EQRoute, in this section. The main idea of our approach is to utilize the estimated available capacity C_j and the sensor demands to automatically determine the maximum layer for constructing the data collection tree. This distributed design also supports collaborative data collection with multiple mobile users. We present our design with two components: a) Construction of data collection tree and b) Migration of data collection tree.

A. Construction of Data Collection Tree

We consider that each sensor holds high priority and low priority data with probabilities p_H and p_L, where $p_H + p_L = 1$. Their information values are denoted by w_H and w_L respectively, where $w_H > w_L$. Similar to most of the studies, the construction of a data collection tree starts with a HELLO message from the mobile user. However, unlike existing approaches, we do not flood the whole network or broadcast to a predefined hopcount. Instead, each node decides whether to extend the tree to the next layer by checking its remaining capacity in a distributed manner.

In our algorithm, the capacity of mobile j, C_j, has to be updated according to its moving speed for energy-efficient and smooth data transmission. It is much easier if we know the coordinates of each sensor and the trajectory of the mobile user. However, we do not make these assumptions, since we want to give more freedom and flexibility for the mobile users to explore new areas. To handle the unpredictable mobility, we introduce ΔD for the mobile user to estimate its valid moving distance from setting up a connection with a new sensor until its disconnection. We pick ΔD as the communication range R in this work to ensure that the tree is updated before mobile user losing connection with its neighbouring nodes. Then, we estimate the available capacity of the mobile user in the time interval $\Delta t = \frac{\Delta D}{v_j}$ by

$$C_j = \frac{\mu_j \Delta D}{v_j}, \tag{5}$$

where μ_j is the service rate and v_j is the moving speed of j.

The mobile user j estimates its available capacity according to its moving speed from time to time. It begins the data collection process by running the CapacityAllocation(j, C_j) algorithm (see Algorithm 2). It first broadcasts to its one-hop neighbours to obtain their capacity requests including the size of high and low priority data, d_i^H and d_i^L, respectively. Then, it assigns capacity first to the high priority data according to the received requests. The neighbouring nodes start transmitting the high priority data immediately after the capacity is allocated. Next, the mobile user assigns the remaining capacity f_i to the low priority data of its neighbouring nodes. If there is still remaining capacity, the mobile user will assign the capacity evenly to its $N(j)$ neighbours, i.e. $f_i = f_j/N(j)$. The neighbouring nodes will extend the tree T_j to the next layer by running the CapacityAllocation procedure. Similar to j, each node i broadcasts to its one-hop neighbours m to receive replies of d_m^H and d_m^L. The capacity allocation process is repeated until there is no remaining capacity left in the data collection tree. Note that the high priority data can preempt the low priority data in the previous layer if there is not enough capacity in the data collection tree.

Algorithm 2 Distributed Capacity Allocation

d_i^H: required capacity for high priority data from node i;
d_i^L: required capacity for low priority data from node i;
f_j: free capacity of node j; initially $f_j = C_j$;

Procedure CapacityAllocation(j, f_j)
Broadcasts Hello message to 1-hop neighbours;
Each neighbouring node i replies with d_i^H and d_i^L;
{//Allocate capacity for H data}
for each reply from neighbouring nodes i **do**
  if $f_j + d_j^L > 0$ **then**
    Allocate capacity for d_i^H;
    $f_j = f_j - \min(f_j + d_j^L, d_i^H)$;
  end if
end for
{//Allocate capacity for L data}
for each d_i^L **do**
  if $f_j > 0$ **then**
    Allocate capacity for d_i^L;
    $f_j = f_j - \min(f_j, d_i^L)$;
  end if
end for
Assign remaining capacity to each i by $f_i = f_j/N(j)$;
{//Extending the tree}
for each neighbouring nodes i **do**
  if $f_i > 0$ **then**
    Run **CapacityAllocate**(i, f_i);
  end if
end for
End Procedure

B. Migration of Data Collection Tree

As discussed before, the mobile user estimates the new capacity of the data collection tree every Δt. However, unexpected disconnections may still occur between the mobile user and the sensors due to its changing speed and moving direction. Hence, the mobile user broadcasts a "MobileHere" maintenance message periodically to its neighbouring sensors to notify them of its existence. In general, the sensors wait passively for the maintenance message. However, they can also check actively for the existence of the mobile user if they do not receive any maintenance messages.

In addition, the MobileHere message can be used for updating the tree structure according to the new location of the mobile user. For example, node i may observe that the mobile user is getting very close if it can receive the maintenance message directly from the mobile user. Node i

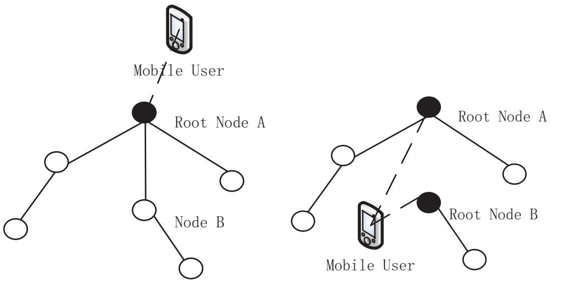
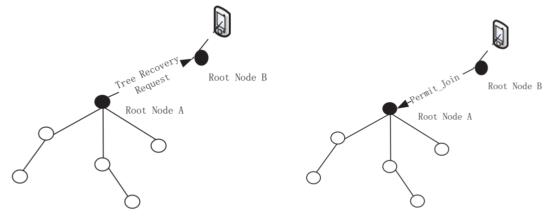
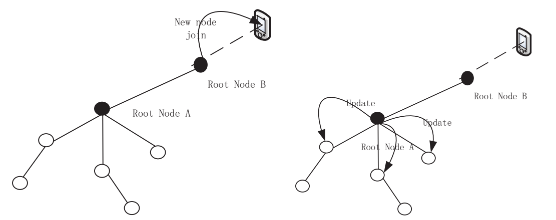


Fig. 2: The data collection tree is reconstructed when the mobile user is moving from the root node to another node inside the tree.



(a) A sends tree recovery request (b) B permits the joining of A



(c) B notifies the mobile user to update the capacity (d) A updates capacity information to its children

Fig. 3: The data collection tree is recovered when the root node detects a disconnection with the mobile user.

can then connect directly to mobile user rather than taking a longer path via a relay node. This scenario can be handled formally by a tree migration process. For better illustration, we divide these migration processes into two types, namely *inner-tree migration* and *tree recovery*.

Figure 2 demonstrates an example of inner-tree migration. At the beginning, the mobile user is connected only to root node A in the data collection tree. Then, it moves to a new location, where it can communicate directly with some other sensors. When sensor node B receives the tree maintenance message from the mobile user, it knows that the mobile user is nearby. Then, node B becomes the root of its subtree and connects directly to the mobile user. After updating the route, node B notifies its previous relay node A and the mobile user to update the capacity assignment accordingly.

Figure 3 shows the tree recovery process. Once root node A detects a disconnection with the mobile, it sends out a "FindMobile" tree recovery message to its neighbours and tries to recover the connection. Any nodes which do not belong to the subtree of A can help relaying the message to the mobile user. In this example, node B relays the message for node A, so that A is reconnected to the mobile user. Similar to inner-tree migration, it is necessary to update the capacity accordingly after tree recovery. Otherwise, the mobile user may think that A has finished transmitting data, while A is still in the tree and has more to send. If the tree recovery process fails, node A may join the data collection trees of other mobile users if they have free capacity. To avoid routing loop, we include the root ID of the subtree that is connecting directly to the mobile user in the tree recovery message. Only the nodes with different subtree IDs will response to the tree recovery request.

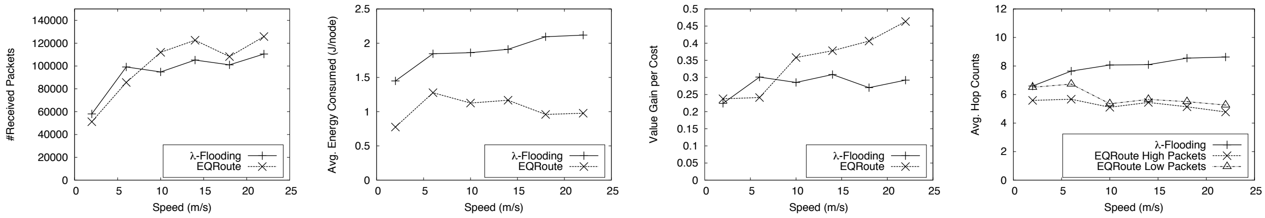
VI. Simulation Evaluation

We evaluate the performance of our distributed algorithm, EQRoute, in OMNet++ simulator [18]. The sensors and the mobile devices communicate with IEEE 802.15.4 in non-beacon mode. The radio operates in the 2.4GHz band with data rate of $250kbit/s$. There are 100 sensors uniformly deployed in a $1000m \times 1000m$ sensing field. The communication range R of the wireless sensors is set to $100m$. We set $\Delta D = 100m$, which is the same as R in our experiments. The mobile users move independently following the random waypoint model, but we do not include any pause times to model continuous and uncontrolled mobility.

We evaluate EQRoute with $p_H = 0.3$, $p_L = 0.7$, and a single mobile user. The data generation rate of the sensors is set to $8B/s$. We vary the mean moving speed of the mobile user from $2m/s$ to $22m/s$ in our experiments with a standard deviation of $0.5m/s$. We compare our EQRoute algorithm with a recently proposed λ-Flooding algorithm [4] for ubiquitous data collection. In λ-Flooding, the mobile user builds a global data collection tree and updates the tree according to a predefined threshold λ to reduce energy consumption in data collection.

Figure 4a shows the total number of received packets from EQRoute and λ-Flooding. We can see that both algorithms receive comparable number of packets. However, our EQRoute algorithm consumes only half of the energy compared with λ-Flooding as shown in Figure 4b. We believe that the energy consumption increases in λ-Flooding due to its prolonged paths to keep the connection with the mobile user. This is further verified by the increasing average hopcount in λ-Flooding as the speed increases (see Figure 4d).

Figure 4c shows the average information value per communication cost (hopcount) of the collected data. EQRoute can achieve much higher information value per communication cost than λ-Flooding, since the highly valued data are given higher priority for collection. We also find that the information value per cost increases with the speed in EQRoute. This is because the mobile user builds smaller data collection trees with small hopcounts when it is moving fast.



(a) Total number of received packets (b) Avg. energy consumption (c) Information value per cost (d) Avg. hopcount

Fig. 4: Simulation results with single mobile user

VII. Conclusions and Future Work

In this paper, we proposed an information-centric ubiquitous data collection algorithm, called EQRoute, for multiple mobile users in wireless sensor networks. EQRoute is a distributed algorithm that allows mobile users to construct the data collection trees dynamically according to their moving speeds. It can control the size of the data collection trees adaptively to reduce packets losses. EQRoute obtains high information value by prioritizing the collection of data carrying more important information. It supports mobile users to make local decisions on data collection without sending any coordination messages among them. Simulation results demonstrated that our distributed solution can improve information value up to 50% and reduce energy consumption by 50% compared with the latest approach.

For future work, we would like to evaluate the distributed algorithm in a sensor network testbed with mobile users. We also plan to exploit mobility prediction to further improve the performance of the distributed algorithm.

VIII. Acknowledgement

This work has been funded by the Swedish Foundation for Strategic Research (SSF) through the ProFuN project on programming platform for future wireless sensor networks. J. Liu's research has been supported by a Canada NSERC Discovery Grant, an NSERC Strategic Project Grant, and a China NSFC Major Program of International Cooperation Grant (61120106008).

References

[1] 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. of IPSN*, 2010.

[2] V. Dyo *et al.*, "Evolution and sustainability of a wildlife monitoring sensor network," in *Proc. of Sensys*, 2010, pp. 127–140.

[3] Y. Kim, T. Schmid, Z. M. Charbiwala, J. Friedman, and M. B. Srivastava, "NAWMS: nonintrusive autonomous water monitoring system," in *Proc. of Sensys*, 2008, pp. 309–322.

[4] Z. Li, Y. Liu, M. Li, J. Wang, and Z. Cao, "Exploiting ubiquitous data collection for mobile users in wireless sensor networks," *IEEE Trans. on Parallel and Distributed Systems*, 2012.

[5] E. C.-H. Ngai, M. B. Srivastava, and J. Liu, "Context-aware sensor data dissemination for mobile users in remote areas," in *Proc. of INFOCOM*, 2012, pp. 2711–2715.

[6] X. Tong and E. C.-H. Ngai, "A ubiquitous publish/subscribe platform for wireless sensor networks with mobile mules," in *Proc. of DCOSS*, 2012, pp. 99–108.

[7] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proc. of MobiSys*, 2009, pp. 55–68.

[8] J. Luo and J.-P. Hubaux, "Joint sink mobility and routing to maximize the lifetime of wireless sensor networks: the case of constrained mobility," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 871–884, Jun. 2010.

[9] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling a three-tier architecture for sparse sensor networks," in *Proc. of IEEE Workshop on Senosr Network Protocols and Applications (SNPA)*, May 2003, pp. 30 – 41.

[10] H. S. Kim, T. F. Abdelzaher, and W. H. Kwon, "Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks," in *Proc. of Sensys*, 2003, pp. 193–204.

[11] Y. Gu, D. Bozdag, E. Ekici, F. Ozguner, and C.-G. Lee, "Partitioning-based mobile element scheduling in wireless sensor networks," in *Proc. of SECON*, Sep 2005, pp. 386–395.

[12] N. Bisnik, A. Abouzeid, and V. Isler, "Stochastic event capture using mobile sensors subject to a quality metric," in *Proc. of ACM MobiCom*, Sep 2006, pp. 98 – 109.

[13] X. Xu, J. Luo, and Q. Zhang, "Delay tolerant event collection in sensor networks with mobile sink," in *Proc. of INFOCOM*, Mar 2010, pp. 1–9.

[14] L. He, Z. Yang, J. Pan, L. Cai, and J. Xu, "Evaluating service disciplines for mobile elements in wireless ad hoc sensor networks," in *Proc. of INFOCOM*, 2012, pp. 576–584.

[15] B. Kusy, H. Lee, M. Wicke, N. Milosavljevic, and L. Guibas, "Predictive QoS routing to mobile sinks in wireless sensor networks," in *Proc. of IPSN*, 2009, pp. 109–120.

[16] H. Lee, M. Wicke, B. Kusy, O. Gnawali, and L. Guibas, "Data stashing: energy-efficient information delivery to mobile sinks through trajectory prediction," in *Proc. of IPSN*, 2010, pp. 291–302.

[17] J. Kho, A. Rogers, and N. R. Jennings, "Decentralized control of adaptive sampling in wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 5, no. 3, pp. 19:1–19:35, Jun. 2009.

[18] A. Köpke *et al.*, "Simulating wireless and mobile networks in OMNeT++ the MiXiM vision," in *Proc. of Simutools*, 2008, pp. 71:1–71:8.