Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 230198, 13 pages
doi:10.1155/2012/230198

*Research Article*

# Data Gathering in Opportunistic Wireless Sensor Networks

**Yongxuan Lai[1] and Ziyu Lin[2]**

[1] *Department of Software Engineering, Xiamen University, Xiamen 361005, Fujian, China*
[2] *Department of Computer Science, Xiamen University, Xiamen 361005, Fujian, China*

Correspondence should be addressed to Ziyu Lin, ziyulin@xmu.edu.cn

Received 23 July 2012; Revised 24 September 2012; Accepted 2 October 2012

Academic Editor: Ruchuan Wang



The wireless sensor networks and opportunistic networks have nowadays presented a trend of technology convergence. On one hand, the nodes periodically sense the environment and continuously generate sensing data; on the other hand, the movements and sparse deployment of nodes usually lead to intermitted connected links and create some form of opportunistic communications. So it is a challenging problem to effectively collect the sensing data in opportunistic wireless sensor networks. In this paper, we propose an efficient data gathering algorithm based on location prediction in opportunistic wireless sensor networks. The algorithm first collects the network metadata such as history of node encounters and contact durations; then it creates a node contact graph, based on which predictive optimal data gathering locations are dynamically calculated and updated. Finally, the sink is controlled to move to these locations to collect sensing data, avoiding lots of unnecessary data exchanges and message transmissions. Extensive experimental results show that the proposed algorithm is effective to reduce the message transmissions and improve the data collection coverage rate.

## 1. Introduction

The wireless sensor networks (WSNs) and the opportunistic networks (or mobile delay tolerant networks) have nowadays presented a trend of convergence [1, 2]. On one hand, the sensor network has emerged some "opportunistic" characteristics in its nature: (1) often deployed in harsh environments, the signal is susceptible to external interference and leads to loss of messages; (2) the node is resource-constrained; many applications would take initiative to turn off the wireless radio devices based on energy considerations, resulting a disconnected network; (3) in mobile sensor networks, the movement of nodes also leads to opportunistic communications; (4) in sparse sensor networks, a mobile sink node is used for message and data collection; the communication is also opportunistic [3, 4]. On the other hand, similar to the "data-centric" sensor network, the opportunistic network is also a "data-related" network. End-to-end paths are not available in the opportunistic network, and it usually adopts a "store-carry-forward" mechanism to forward messages [1, 5]. The content of packets being forwarded plays an important role in the routing scheme. So the opportunistic wireless sensor network (OWSN) is a kind of opportunistic network that consists of nodes with sensing capabilities and actively or passively adopts the form of opportunistic transmissions. Research on OWSNs is a result of technology integration between wireless sensor networks and opportunistic networks.

Data gathering is a key application in the "data-centric" sensor networks. For example, in agricultural applications the deployed sensor network would collect the sensed data such as temperature and humidity for analysis [6]; in habit monitoring applications, data such as occurrence, skin color, and locations of the animals are to be collected [7]; recently Ayaki et al. [8] proposed a data gathering scheme in urban streets using mobile phones as the relayed nodes. However, the "opportunistic" characteristic makes the data gathering a challenging problem in OWSNs. Firstly, static communication links (or paths) between the source node and the sink are not available in the opportunistic network. Nodes adopt a "store-carry-forward" mechanism for message transmissions, and most of the existing data gathering algorithms based on a stable link assumption in WSNs cannot work properly. Secondly, most of the research in opportunistic network focuses on the message diffusion rather than particularly on the issue of data gathering. The goal of message

diffusion algorithm is populating the messages to their varied destinations efficiently; yet for the data gathering algorithm, there is usually only one target (the sink node) for the large amount of sensed data that are injected into the network from different sources. So existing routing schemes cannot work properly for the data gathering applications in OWSNs.

In this paper we study the problem of data gathering in the OWSNs and propose an efficient data gathering algorithm called PDA (predictive dynamic data-gathering algorithm). The idea behind PDA is that nodes in OWSNs, for example, mobile phones, do not usually move randomly, but would roughly move according to some relatively stable patterns. If the moving pattern is known to the sink, the sink would then be programmed to move to the optimized locations to collect the sensing data, and lots of message exchanges and transmissions would be avoided to enhance the performance. More concretely, our algorithm firstly collects the network metadata such as the history of node encounters, contact durations, and so forth for modeling. Then it generates a node contact graph, based on which the predictive optimal locations are calculated. Finally, the sink is controlled to move to these locations to collect the sensing data and avoid lots of transmissions. Compared with other data gathering algorithms, PDA has the following features and advantages.

(1) Data transmissions are based on the pattern of node movements. The algorithm collects the metadata for modeling to predict the optimized data gathering points, and it selectively exchanges data among the nodes, which cuts down the transmissions of data gathering.

(2) The algorithm utilizes abundant resources of the sink node. The sink is programmed to travel among several optimized data gathering points to collect the sensing data, which avoids unnecessary data transmissions among ordinary nodes.

(3) The algorithm attaches great importance on the success of "last hop" transmissions to the sink. Nodes with large volume of sensing data from other nodes stored on their storage would have larger probability to encounter with the sink and have enough connection length to route their data to the sink.

Extensive experimental results show that the proposed algorithm can reduce about 40∼60 percent of message transmissions and improve data collection coverage rate about 8∼12 percent compared with other epidemic and probabilistic data gathering algorithms.

The rest of the paper is structured as follows. Section 2 surveys some existing research related to this paper. Section 3 gives some assumptions about the network model and defines the cost model of data gathering. Section 4 describes the detailed mechanism of the proposed scheme PDA, including the initial run, calculation of data gathering points, data collection, and update of data gathering points. Finally, Section 5 describes the experimental setup and performance evaluation, and Section 6 concludes the paper.

## 2. Related Work

There exists some research on the data gathering applications in traditional MANET (mobile ad hoc network) and sensor networks, yet they assume the nodes are fixed and have stable links for communication [9, 10]. For the opportunistic network, most of the existing research focuses on the forwarding and routing of messages; there is no much work on the research of data gathering [5, 6, 11]. So in this section we survey some related work from areas of data forwarding and data gathering in the opportunistic networks and sensor networks.

*2.1. Data Forwarding.* Opportunistic networks adopt a "Store-Carry-Forwarding" strategy for message transmission [5, 11]. Messages are temporally stored on the nodes, and if there exists chance of communication at some proper time, the messages are forwarded to other nodes and finally routed to their target nodes. So utilization of the chance of opportunistic channel, constraint of energy and storage space, and the rate of successful message transmissions are all key factors for the message routing and forwarding algorithms [12].

Epidemic routing [13] uses a flooding-like mechanism for message routing. Encountered nodes would fully take advantage of the communication chance for message exchange to increase the rate of successful message transmission and decrease the message delay. To cut down message transmissions and handle the network congestion, controlled-flooding algorithm [14] is proposed, which selectively forwards the message based on the forwarding probability, as well as time-to-live (TTL) or kill time. In the prioritized epidemic routing PREP [15], messages are forwarded or deleted based on the current cost to destination, current cost from source, expiry time, and generation time, and messages closer to the destinations would have more copies to improve the rate of successful transmission.

Besides the flooding-like scheme, another type of message forwarding scheme is to utilize context information and knowledge of the network to optimize the message transmissions. ZebraNet project [7] uses a mechanism based on the history of node movements. Each node maintains a probability to the sink, and the node with higher probability would send its messages to the node with lower probability when two nodes encounter. Similar to [7], PROPHET [16] computes a forwarding probability based on the historical record of its observed contacts, and messages are routed to its neighbor only if it has higher probability than the neighbor. The CAR (context-aware routing) scheme [17] takes the contextual factors such as residual energy, change rate of topology, and moving speed as the input and uses Kalman filter to calculate the probability to the destination node. Messages are sent to the node with the highest probability. Most of the schemes mentioned above assume a random mobility model where the chance for communication is usually by accident and uncertain.

More recently, studies have focused on mobile social networks (special type of DTNs consisting of human-carried devices) and analyzed the social network properties of these networks to assist the design of routing algorithms, where

data forwarding metric is centrality-based. bubble rap [18] uses betweenness as the centrality metric which measures the social importance of a node facilitating the communication among other nodes. In *friendship-based routing* [19], friendships in terms of their behaviors are defined between nodes (i.e., people) to facilitate message forwarding. In addition [20] studies the transient characteristics of node contact patterns. It formulates the transient social contact patterns of mobile nodes as a Gaussian function, based on which it develops data forwarding metrics to analytically predict the contact capability of mobile nodes with better accuracy. In addition [21] proposes a community-based message forwarding scheme CMTS. It organizes the network into different communities based on the contact frequencies among them. The number of message copies is determined by the communities, and messages are transmitted to target communities mainly relying on active nodes that have larger social degrees.

Yet schemes mentioned above are general message transmission schemes, which are not optimized for the data gathering application. The algorithm proposed in this paper would learn the moving pattern of nodes and actuate the sink node properly to improve the overall performance.

*2.2. Data Gathering.* Nodes in sensor networks continuously sense the environment and generate large volume of data. Traditional algorithms such as TAG [10], PULL/PUSH [9] all depend on infrastructures such as query trees or clustering to collect data. Yet in OWSNs these infrastructures are expensive to maintain as the full point-to-point links are usually not available, so *infrastructure-free* strategies are adopted in the data gathering algorithms.

Reference [6] introduces a real deployment of opportunistic network in agricultural applications which uses the PBR (potential-based routing) algorithm for data gathering. By setting higher potential to remote sensors and lower potential to data server, sensor data is autonomously gathered at lower potential nodes. In data mules system [22], the mule nodes collect the sensing data and route them to the access point through one or multiple hops of transmissions. In the *message ferrying* mechanism [23], ferry node loads data from the source nodes and then forwards them to the middle nodes while moving along its path. The path could be predefined or resolved from the requests of the ordinary nodes. Reference [24] uses a predictable mobility model to collect sensing data, where nodes would learn the connection time with the mobile nodes (e.g., buses) and wake up at that time to facilitate data communication. Our algorithm also learns the mobility model to improve the performance, but we assume all nodes are mobile nodes and besides the connection time, our scheme introduces other factors such as location, contact duration, and so forth for optimization.

Reference [4] uses a mobile sink node to collect static sensor readings by controlling the movement of the mobile sink. It models the data collection problem as sensor selection problem and analyzes the design of a feasible movement route for mobile sink to collect data and minimize the velocity requirements. The schemes use mobile sink node to assist in data gathering, but it assumes the ordinary nodes are static. In OWSNs, all the nodes are mobile and they adopt a style of opportunistic communication, so we need new and energy efficient algorithms that are suitable for the network to collect the sensing data.

## 3. Network Model and Cost Model

In this section, we define the network and analyze the cost model for data gathering within the network.

*3.1. Network Model.* We assume the network has the following characteristics (as in Figure 1):

(1) there are one sink node and $N$ sensing nodes, every node has a unique id $s_i$;

(2) nodes periodically sense the environment and generate a tuple. These tuples are to be gathered at the sink; all copies of the same message have the same message id;

(3) the sensed nodes have constraint on energy and storage space, yet the sink has no limit on these aspects;

(4) nodes have the knowledge of their locations, and they could record their locations at proper time;

(5) the movement displays some kind of patterns in cycles; and the sink is programmed and actuated by applications.

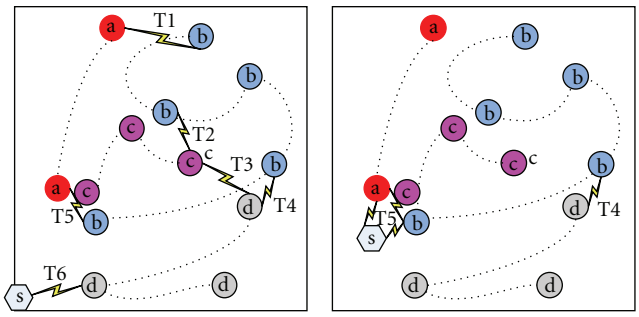
The network uses an opportunistic way for message transmission. When two nodes encounter, they establish a temporary communication link for message exchange. Here we define some notions of network connection.

*Definition 1* (chance for communication). Given $L_{i,j}(t)$ is a period of time when node $s_i$ and $s_j$ are within each other's radio range in cycle $t$, if the length of $L_{i,j}(t)$ is larger than a predetermined threshold value ($L_{i,j}(t) \geq \phi$), then there exists a *chance for communication* between the two nodes in cycle $t$, denoted as $s_i - s_j(\phi, t)$.

For the sake of convenience, we use $L_{i,j}$ to denote the connection length within the current cycle, and we use $s_i - s_j$ to denote there exists a chance for communication between node $s_i$ and $s_j$ in the current cycle.

*Definition 2* (path). For node $s_i$ and $s_j$, if there exists a collection of nodes $\{m_1, m_2, \ldots, m_{k-1}\}$ such that there are communication chances $s_i - m_1, m_1 - m_2, \ldots, m_{k-1} - s_j$,then there exists a path from $s_i$ to $s_j$, denoted as $\text{path}_{ij}$. In addition $|\text{path}_{ij}| = k$ indicates the length of the path.

*3.2. Cost Model of Data Gathering.* As the network consists of $N$ sensing nodes and one sink node, the collection of all encounters could be denoted as $\Omega = \{(t_k, s_i, s_j) \mid t_k \in \{1, \ldots, \text{total}\}, s_i, s_j \in \{s_1, \ldots, s_n, \text{sink}\}\}$, where $(t_k, s_i, s_j)$ denotes that node $s_i$ and $s_j$ have a communication chance at time $t_k$, and $\{1, \ldots, \text{total}\}$ is the time span under observation. Based on $\Omega$, the data exchange collection could be defined as




(a) Probability-based (b) PDA

Figure 1: Illustration of data gathering strategies in OWSN.

$\Phi = \{(t_k, d_{ij}) \mid t_k \in \{1, \ldots, \text{total}\}, (t_k, s_i, s_j) \in \Omega\}$, where $(t_k, d_{ij})$ denotes that node $s_i$ sends data $d_{ij}$ to $s_j$ at time $t_k$. According to these two collections, the cost of data gathering for the network is

$$\text{Cost} = \sum \left| d_{ij} \right| * T_{ij}, \quad \left(t, d_{ij}\right) \in \Phi, \tag{1}$$

where $|d_{ij}|$ denotes that the size of packets of data $d_{ij}$, $T_{ij}$ is the cost of transmitting unit data from node $s_i$ to $s_j$. The data gathered at the sink is

$$D = \sum_i |d_{is}|, \quad (t, d_{is}) \in \Phi, \tag{2}$$

where $s$ denotes the sink. If the total data $s_i$ generates $d_i$, then the data coverage of the network is

$$\text{Cove} = \frac{D}{\sum_{i=0}^{n} |d_i|}. \tag{3}$$

Since the sink could be programmed and actuated by applications, the time and place that the sink encounters with other nodes belong to the calculation of the data gathering problem. So the goal of the data gathering algorithm is to calculate the encounter collection $\Omega$ and data exchange collection $\Phi$ to minimize the total message transmissions while the data coverage Cove is above certain threshold $\rho$. At the same time, the node $s_i$ and $s_j$ should have enough bandwidth to transmit the exchanged data $d_{ij}$. Therefore, the formalization of data collection could be defined as

$$\begin{aligned} \text{minimize}: \ \text{Cost s.t.} \quad & \text{Cove} \geq \rho, \\ & d_{ij} \leq \text{BW}\left(s_i, s_j\right)^* L_{i,j}(t_k), \\ & \left(t_k, d_{ij}\right) \in \Phi, \qquad \left(t_k, s_i, s_j\right) \in \Omega, \end{aligned} \tag{4}$$

where $\text{BW}(s_i, s_j)$ is the bandwidth of the link between node $s_i$ and $s_j$. From (4) we could see that when the network topology is fixed, the optimized data collection could be transformed into the *minimum-cost flow problem* [25] if a virtual source node is added to the network and connects to every sensor node in the network (see The Appendix). But in opportunistic networks, no centralized node is available for computing, and the network topology changes dynamically due to the moving nodes. When a node moves, the current optimal solution for data gathering becomes invalid. So the resolution of (4) is meaningful only in cases when nodes move according to some kind of patterns, for example, office workers traveling among home, bus stations, and offices at roughly the same time every day.

## 4. Data Gathering Algorithm Based on Location Prediction

*4.1. Algorithm Overview.* This section describes details of the proposed data algorithm PDA (predictive dynamic data-gathering algorithm) in OWSNs. As an overview, Figure 1 compares the PDA strategy with a probability-based data gathering scheme in OWSNs. The circles denote the nodes and the dashed lines denote the paths along which the nodes move, and Table 1 presents the time, nodes, and duration of the connections. In Figure 1(a), the sink $s$ encounters node $d$, and $d$ encounters nodes $b$ and $c$, so the data forward probability is $d > b = c > a$. According to the probability-based data forwarding algorithms (e.g., PROPHET [16]), at T1 node $a$ would forward its data to $b$; at T2 nodes $b$ and $c$ would exchange their data; at T3 and T4 nodes $c$ and $d$ would route their data to $d$; at T5 nodes $a$, $b$, and $c$ would exchange their data; finally at T6, after gathering data from other nodes, node $d$ encounters the sink, yet it has only 0.5 s to route all the data to the sink. Part of the data cannot be gathered successfully by the sink, causing data loss in the application. Instead, if the nodes move according to some relatively stable patterns, after some necessary data exchanges, the sink would learn the pattern of node movements and move to the data gathering point to collect the data at T5, which cuts down several hops of transmissions and has enough time to collect the sensing data (Figure 1(b)). Moreover, as nodes $a$, $b$, and $c$ have stable connection with the sink at T5 (10 s), PDA suppresses data transmissions between nodes $a$ and $b$ at T1 and between $b$ and $c$ at T2, which further cuts down the data transmissions.

There are 4 phases in the PDA algorithm: (1) the sink collects metadata such as historical node movements, the contact durations, and so forth for modeling; (2) the sink builds an encounter graph to dynamically compute the optimal locations of data gathering points; (3) the sink is controlled to move to data gathering points to collect the sensing data, saving lots of transmissions and improving the data collection coverage; (4) gathering points are updated if necessary.

*4.2. Initial Run.* At the very beginning, PDA adopts the epidemic routing strategy [13] for message transmission and data gathering. The sink moves according to the way-point mobility model for information exchange in the sensing field. Each node, for example, $s_i$, would send its sensing data and moving history to the sink when it encounters the sink. The data of moving history is represented and stored in *MTable* (*movement table*), which includes the timestamp, location,

Table 1: Encountered nodes and their contact duration.

| Time | Nodes and flow of data | Duration (s) |
|---|---|---|
| T1 | $a \rightarrow b$ | 1 |
| T2 | $b \rightarrow c$ | 3 |
| T3 | $c \rightarrow d$ | 4 |
| T4 | $b \rightarrow d$ | 3 |
| T5 | $a \rightarrow b$, $a \rightarrow c$ | 10 |
| T6 | $d \rightarrow s$ | 0.5 |

node id, contact duration, and so forth. Table 2(a) shows an example of *MTable*.

The length of initial run is 1 sliding window. Within the time span *MTables* are disseminated from the ordinary nodes to the sink and also among the ordinary nodes. To avoid the over-flooding problem, PDA uses filtered and compressed version of *MTables*. For example, Table 2(b) shows a compressed version of the *MTable* in Table 2(a), where the compressed *MTable* only includes tuples whose contact duration is greater than 1.0s. Controlled-flooding strategies are also used to cut down transmissions in this phase, which would be further discussed at Section 4.4.

### 4.3. Calculation of Data Gathering Points

*4.3.1. Constructing the Contact Graph.* The sink would merge *MTables* from ordinary nodes and use them as the input for constructing the contact graph. Firstly, for every node $s_i$ in the merged *MTable*, PDA calculates its encounter probability $M_{ij}$ and total connection length $\mathrm{TL}_{ij}$ with node $s_j$:

$$M_{ij} = \sum_{K=1}^{T-1} f(k),$$
$$\mathrm{TL}_{ij} = \frac{1}{T-1} \sum_{K=1}^{T-1} L(i,j)(k), \quad T > 2, \tag{5}$$

where $T$ is the current epoch, and $\{1, 2, \ldots, k, \ldots, T-1\}$ consists of a sliding window with $(T-1)$ epochs. The encounter probability and total connectivity length are all calculated within this window. $f(k)$ is a user-defined weighting function, when $s_i$ encounters $s_j$ at $k$th epoch and contact each other (denoted as $s_i - s_j$), $f(k) = 1/2^{T-k}$; otherwise when they do not encounter each other, $f(k) = 0$. $L_{i,j}(k)$ denotes that the connection length of node $s_i$ and $s_j$ at $k$th epoch; when $s_i$ and $s_j$ do not encounter, $L_{i,j}(k) = 0$. As illustrated in the definition of $f(k)$, PDA assumes that later encounter has greater impact on the weight. For example, when $T = 5$, if node $s_i$ encounters $s_j$ at the 2nd, 4th epochs, then the encounter probability $M_{ij} = f(2) + f(4) = 1/2^{5-2} + 1/2^{5-4} = 0.625$.

If encounter probability $M_{ij}$ meets some threshold $\delta$ ($M_{ij} \geq \delta$), then an edge that has $s_i$ and $s_j$ as its vertexes is added into the *contact graph* (CG). Figure 2(a) is an example graph that is transformed from Figure 1. PDA assumes that the nodes are densely deployed so that most of the nodes are included into the connected contact graph. The movements of the isolated nodes not in the graph usually lack patterns, so they have little impact on calculating the data gathering points. It is worth mentioning that Figure 2 only shows the encounter probability $M_{ij}$ on the edges, yet other information such as timestamp, contact duration, and locations is also included in the edges.

Table 2: Examples of *MTable*.

(a) *MTable* of node $s_1$

| Time | Location | Node | Duration ($\times 10^{-1}$ s) |
|---|---|---|---|
| 5 | (30, 30) | $s_2$ | 5 |
| 12 | (30, 30) | $s_3$ | 13 |
| 24 | (50, 40) | $s_4$ | 20 |
| 26 | (50, 40) | $s_5$ | 22 |
| 52 | (70, 80) | $s_2$ | 10 |
| 55 | (80, 80) | $s_6$ | 2 |
| 62 | (80, 90) | $s_7$ | 6 |

(b) Compressed *MTable* of node $s_1$, threshold $\geq 10$

| Time | Location | Node | Duration ($\times 10^{-1}$ s) |
|---|---|---|---|
| 12 | (30, 30) | $s_3$ | 13 |
| 24 | (50, 40) | $s_4$ | 20 |
| 26 | (50, 40) | $s_5$ | 20 |
| 52 | (70, 80) | $s_2$ | 10 |

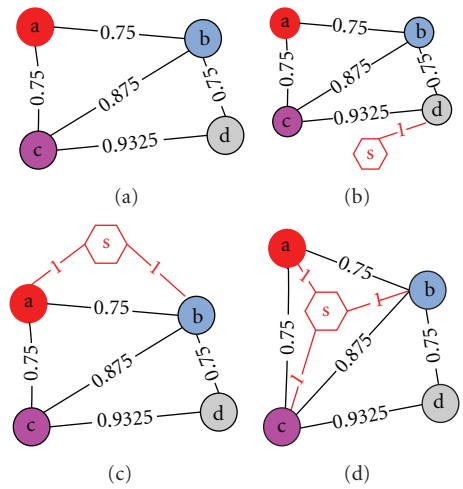



Figure 2: Contact graph $G$ and possible positions of the sink.

*4.3.2. Calculation of Data Gathering Points.* The contact graph is actually a sketched description of the moving pattern of nodes, and it is used to calculate the data gathering points. Then the sink would move to these predefined gathering points to collect the sensing data. Figures 2(b) and 2(c) illustrate the possible positions of the sink when gathering the data.

We denote the position of the sink as $s$ and denote the encounter probability of node $s_i$ and sink as

$$M_{is} = \min\left(1, \frac{L_i}{\mathrm{Th}}\right), \tag{6}$$

where $L_i$ is the connection length between node $s_i$ and sink; Th is the minimal time interval for a connection. Also we denote the contact graph as $G'$ when $s$ is added to $G$, and we can construct a spanning tree routed at $s$ from $G'$. Data is then transmitted from the internal nodes to the root of the tree. Suppose the cost of transmitting unit data between any two nodes is $T_{ij} = 1$, then the data gathering cost Cost and expected amount of gathered data $D$ defined in (1) and (2) in Section 3.2 could be transformed to

$$\begin{aligned} \text{Cost}(s, G) &= \sum_a |\,\text{path}(a, s)\,|, \quad a \in \text{Tree}, \\ D(s, G)_e &= \sum_a |d_a| * \Pi M_{ij}, \quad i - j \in \text{path}_{as},\ a \in \text{Tree}, \end{aligned} \tag{7}$$

where $\text{path}_{as}$ is the path from node $a$ to the sink $s$in the spanning tree $|\text{path}_{as}|$ is the hops $i - j$ is the edge within the path; $d_a$ is the data generated by $a$; $\Pi M_{ij}\,(i - j \in \text{path}_{as})$ is the probability of data gathered from node $a$ to the sink. The optimal position $s^*$ for sink in the graph $G'$ meets the following conditions:

$$\text{Cost}(s^*, G) = \text{Minimal}(\text{Cost}(s, G)), \tag{8}$$

$$D(s^*, G)_e \geq \rho^* \sum_a |d_a|, \tag{9}$$

where $\rho$ is the predefined threshold of the data coverage. When nodes move randomly, it is undesired to calculate the optimal data gathering positions; when nodes move according to some kind of pattern, then the edges in the contact graph represents larger chance of encounters and longer connections among the nodes. So the optimal data gathering points could be calculated according to (9).

A direct method for calculating $s^*$ in (9) consists of two steps: (1) construct an extended contact graph $G'$ for each stable edge by adding the sink linking to the vertexes into the graph $G$; (2) for each $G'$, generate the spanning tree who has the minimal transmission cost and also satisfies the minimal flow constraints. For example, Figures 3(a) and 3(b) are two generated spanning trees that have the minimal $\text{Cost}(s, G)$. Yet combined with Figure 1(b), node $a$ and $b$ have larger connection length with the sink, so Figure 3(b) is a preferred position for the scheme. Also if more than 2 nodes are connected during the same time period, then the sink could be added to the graph to calculate the spanning tree, which is illustrated in Figure 3(c).

When the data gathering point is decided, suppose the set of vertexes connected with the sink is $P$, then the nodes represented by $P$ could all connect with sink during some time period. The optimal location of the sink is defined as the geometrical center, which could be calculated from the information such as connection timestamp and location contained at edges of the contact graph.

$$s \cdot \text{Interval} = \cap L_{i,j}(t), \quad \forall s_i, s_j \in P, \tag{10}$$

$$s \cdot x = \frac{1}{|P|} \sum a \cdot x, \qquad s \cdot y = \frac{1}{|P|} \sum a \cdot y, \quad a \in P, \tag{11}$$

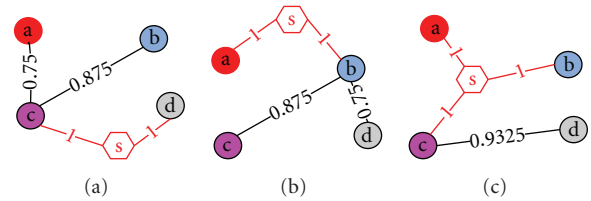



Figure 3: Possible spanning trees when the sink is added to the graph.

where $(a \cdot x, a \cdot y)$ is the location of node $a$; $|P|$ is the number of nodes $P$ contains. Note that the optimal position of the sink $s^*$ could be a set of positions that the sink could travel along to enhance the efficiency of data gathering, and we will discuss it later in Section 5.3.

*4.4. Data Gathering.* When the gathering position is decided, a set of nodes $P_u$ could also be defined, and PDA calculates the *data forwarding probability* (DFP) for the nodes in the set. If $u = 0$, $P_0$ is the sink; if $u = i\,(i \geq 1)$, $P_i$ is the sink and its $i$-hop neighbor in the contact graph. The DFP of the sink is denoted as $F_s = 1.0$; if node $a$ links to node $s$ directly or indirectly, then the DFP of node $a$ is

$$F_a = \Pi M_{ij}, \quad i - j \in \text{path}_{as}, \tag{12}$$

where edge $i - j$ belongs to the path from $a$ to sink. The data forward probability of the nodes within $P_u$ is then flooded to the network, and nodes that do not belong to $P_u$ could deduce their probability according to the *MTable's* when they receive the DFPs from other nodes. The DFP of a node $k \notin P_u$ is initially set to be 0; if node $k$ ever contacts some node $a \in P_u$, then the DFP of node $k$ is

$$F_k = \max(F_a * M_{ak}), \quad a \in P_u,\ k \notin P_u,\ s_a - s_k, \tag{13}$$

where $M_{ak}$ is the encounter probability between node $k$ and $a$. When two nodes move close enough to establish a connection, if their difference on DFP is larger than some predefined threshold, then the data is transmitted from the node with smaller DFP to the node with larger DFP. As shown in Figure 3(c), $P_1$ is the set of nodes $\{s, a, b, c\}$, and their DFP is 1.0; so in Figure 1, node $a$, $b$ and node $b$, $c$ suppress their data transmissions at T1 and T2 as they have the same DFP; then at T5, the nodes move to the radio range of the sink, so they send their data to the sink and finish the data gathering at this epoch.

$P_u$ is diffused within the network, and its size could be adjusted according to the applications. When two nodes encounter, they first exchange $P_u$ and then calculate their DFPs to decide whether a further data transmission is needed. According to (13) and (14), the optimal data gathering point $s^*$ guarantees that the nodes within $P_1$ have larger DFP. These nodes could be viewed as agents of the sink, and they expect to send their data to the sink through 1 hop of transmissions.

*4.5. Update of Data Gathering Points.* Although we assume the pattern of node movements are relatively stable, as time

goes by the gathering points may be obsolete. In more detail, there are two cases when PDA needs to adjust the data gathering points.

(1) The sink encounters most of the nodes in $P_u$, but the data coverage is relatively small. In this case, the sink would recalculate the data gathering point according to the algorithm described in Section 4.3 as nodes also record their *MTable's* and route them to the sink periodically and hence the sink has all the metadata for the calculation.

(2) The sink only encounters a small part of the nodes in $P_u$, and the data coverage is small. For this case, the sink would initiate an *initial run* as described in Section 4.1 to collect the metadata and sensing data using the epidemic routing strategy to update the data gathering points.

(3) It is worth mentioning that the *MTable's* are not affected by the DFP's of the nodes; they are flooded within the network. To cut down the cost of metadata transmissions, a TTL (time-to-live) segment is inserted into the message. When a piece of metadata is copied to a node, TTL increases by 1; the copy is stopped when TTL increases up to the predefined threshold. Moreover, a probability is used for the metadata exchange. When node $s_i$ meets $s_j$, the nodes would exchange the metadata with probability $\gamma < 1$, which disperses the data exchange within the time dimension. So according to the strategies of controlled flooding, sink could collect enough metadata and avoid the problem of overflooding.

## 5. Experimental Study

We implement PDA in C# and compare it with other data gathering schemes. The experimental result shows that PDA reduces about 40~60 percent of message transmissions and improves data collection coverage rate about 8~12 percent, compared with other epidemic and probabilistic data gathering algorithms.

*5.1. Environment Setup.* To simulate the movement pattern of nodes, we adopt the *community model* described in [16]. In the model, the network is divided into $K * K$ grids (also called community, illustrated as Figure 4). Each community has an *interest index* $c_i \in (0, 1)$. If the index is greater than a threshold, the community is called the *hot community*, and the set of hot communities in the network is denoted as $\mathbb{C}$. Similar to [8] nodes move along paths, yet in different ways: (1) node $s_i$ could move along *Max_Path* paths at most; (2) if $p_i$ is a path of node $s_i$, then $p_i$ consists of *Max_Com* communities at most, and the start and the end points are the *Home Community* of the node; (3) along the path there are at least $m$phot communities belonging to $\mathbb{C}_i$, where $\mathbb{C}_i \subseteq \mathbb{C}$ is the set of hot communities of $s_i$. When a node moves along the path and travels on community $C_i$, it would stop for a period with probability of $P_{\text{stop}} = ps + (1 - ps) * c_i$ and would choose to move to the next community along the path with probability of $1 - P_{\text{stop}}$, where $ps$is the predefined value; $c_i$ is the interest index of the target community. $(1 - ps) * c_i$ represents the probability that a node would stop because of the distraction within the community that it currently visits. The simulation runs periodically, a node has several predefined paths, and the path is randomly chosen for the node. Meanwhile, nodes may not travel along their path with probability $\eta$, called the disturbance factor. The simulation consists of 10 rounds, and the length of each round is 100 s. Each node generates 0.2 packets per second, where the size of the packet is 512 B. So the size of sensing data that each node generates and is to be collected by the sink is 25 k (100 s $*$ 0.2 Packet/s $*$ 512 B/Packet) each round. Also, we assume that all *MTable's* could be wrapped into 1 packet.

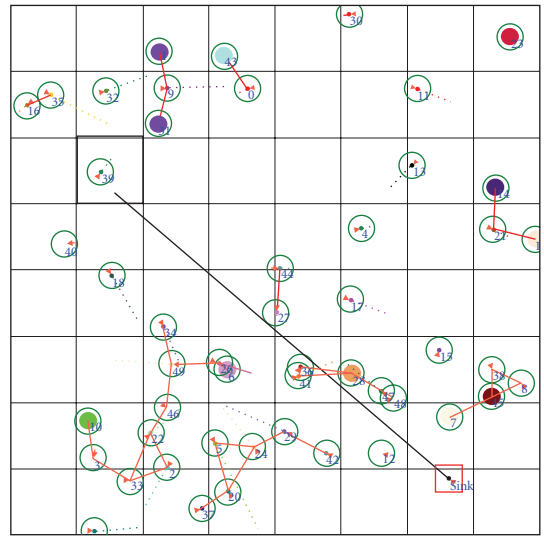



Figure 4: Snapshot of the simulation field ($N = 50$).

Three other data gathering algorithms are compared with our algorithm: (1) Random: the sink moves according to the way-point mobility model and collects data from the nodes it encounters. There is no data exchange between ordinary nodes; (2) Epidemic [13]: the sink and ordinary nodes take advantage of all chances of communications and data are exchanged among any nodes if possible and finally routed to the sink; (3) PROPHET [16]: data are exchanged according to the data forwarding probability which is maintained by the nodes using the movement history. As the cost of communication dominates the depletion of the limited battery energy in sensor nodes, we present only the total communication cost (number of packets) incurred by various algorithms. We assume ideal links when two nodes meet and establish a connection. Table 3 lists the default parameters in the simulations.

*5.2. Performance Comparison and Analysis*

*5.2.1. Impact of Network Size and Radio Range.* From Figures 5 and 6 we would see that the size of the network has great impact on the message transmissions, yet has little impact on the data coverage. Random has the smallest number of

Table 3: Parameters of the simulation.

| Parameter | Value | Description |
|---|---|---|
| $N$ | 50 | Number of nodes |
| $W$ | 120 m | Width and length of the field (120 m $*$ 120 m) |
| $K$ | 8 | Divided into $K*K$ communities |
| Sim_$T$ | 2000 s | Length of simulation |
| $E$ | 100 s | Length of each round |
| $O\_T$ | 5 | Number of rounds in the sliding window |
| Th | 1 s | Minimal length of time for a valid connection |
| $R$ | 6 m | Radio range for a node |
| $\eta$ | 0.1 | Disturbance factor when a node travels along its path |
| st | 10 s | Length when a node stops |
| Min, max | 6, 10 m/s | Min/max speed of nodes |
| $\mathbb{C}$ | 9, 15, 20, 17, 5, 48, 56 | Id set of hot communities |
| mp | 1 | Minimal number of hot communities in the paths |
| Max_Path | 2 | Maximal number of paths a node has |
| Max_Com | 10 | Maximal number of communities a path may travel |
| Max_Copy | 5 | Maximal number of versions a packet may have in Epidemic routing |
| Max_Buffer | 500 | Size of cache buffer (packet) |
| $\delta$ | 0.6 | Threshold of encounter probability when constructing the graph |

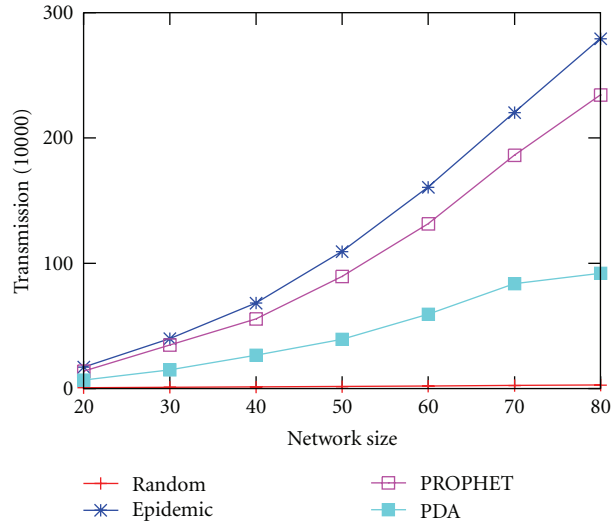



Figure 5: Network size versus messages.

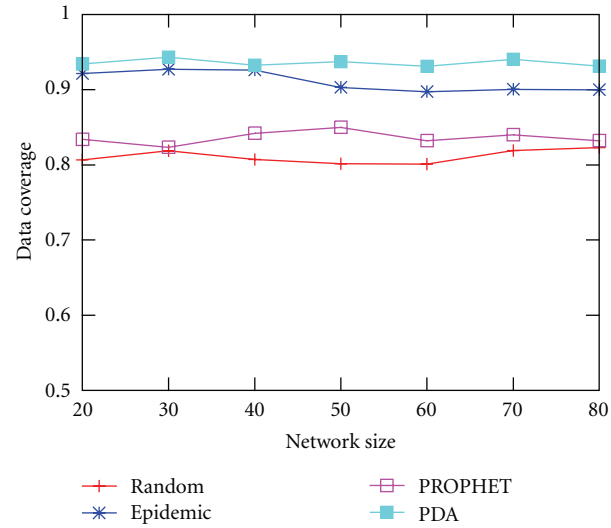



Figure 6: Network size versus data coverage.

transmissions (about 5% of that in Epidemic); Epidemic and PROPHET have the largest number of transmissions, and the performance of PDA is in the middle, which cuts down about 38% of transmissions. But from the view of data coverage, Random has a coverage of 80%, and PDA is about 5~10% higher than Epidemic (about 89%) and PROPHET (about 83%). This lies on two aspects: (1) PDA takes advantage of the pattern of node movements, and controls the sink to collect the data at optimal data gathering points; (2) PDA requires less storage space than other algorithms (discussed in Section 5.2.2). When $N = 50$, the sink in Random moves $2.8*10^3$ m due to its random movement; yet in PDA, the path is only about $0.4*10^3$ m long because the sink only moves among the data gathering points, which partly cuts down the energy consumption.

Figures 7 and 8 shows the impact of radio range. From the figure, we could see that the data transmissions and coverage grow with the radio range. Random has the best performance on transmissions, and PDA has the best performance on data coverage. Yet when radio range grows up to 16 m, the data coverage and transmissions increase to their maximal. Note that when radio range is large (e.g., 24 m), the number of message transmissions in Random is about $1.8*10^4$ and the data coverage is about 93%, which is a good performance for data gathering. Yet considering the increase of radio range is constrained by the hardware and would greatly increase the cost of per-unit data transmissions, to improve the performance by increasing the radio range is actually impractical in real deployments.

*5.2.2. Impact of Buffer Size.* As the algorithms adopt a "Store-Carry-Forward" scheme for message transmissions, the size of buffer would have a great impact on the performance. In Figures 9 and 10, every node would generate 0.2 packets per second, so each node would have 400 ($2000 * 0.2$) packets as its own sensing data. As the figures show, the number of transmissions and data coverage are small when the buffer size is small. Random requires the least buffer as each node only stores its own sensing data in the buffer. Compared with Random, Epidemic and PROPHET have poorer performance when the buffer size is less than 150 packets. This is because

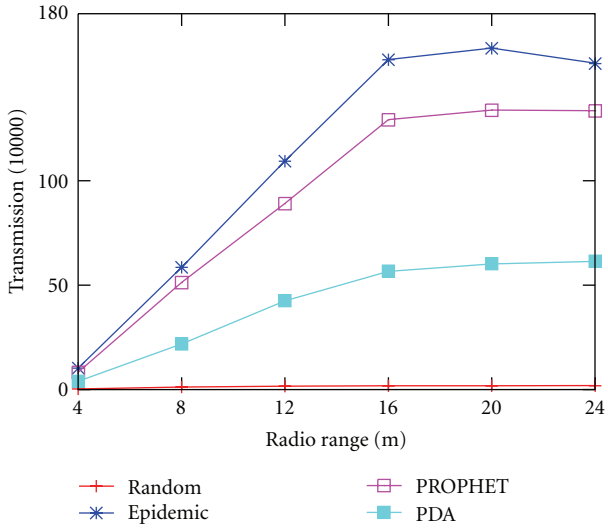



Figure 7: Radio range versus messages.

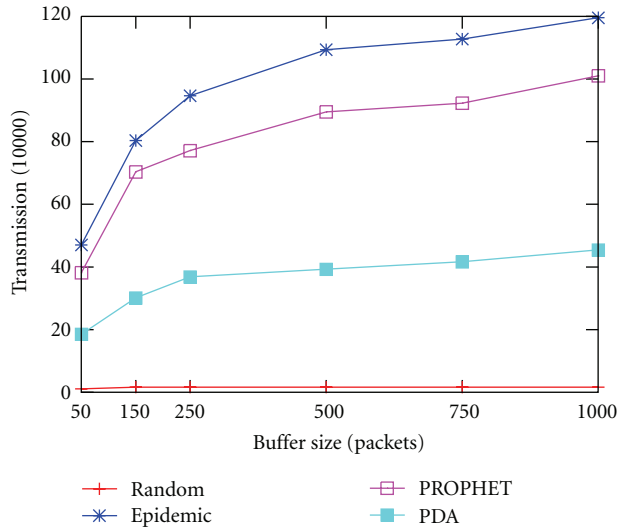



Figure 9: Buffer size versus messages.

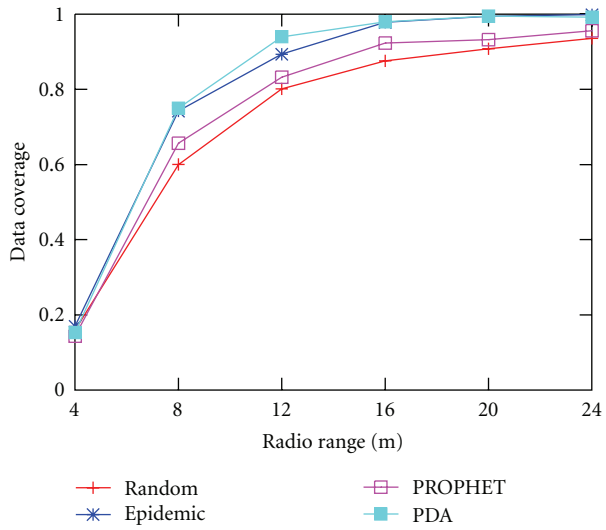



Figure 8: Radio range versus data coverage.

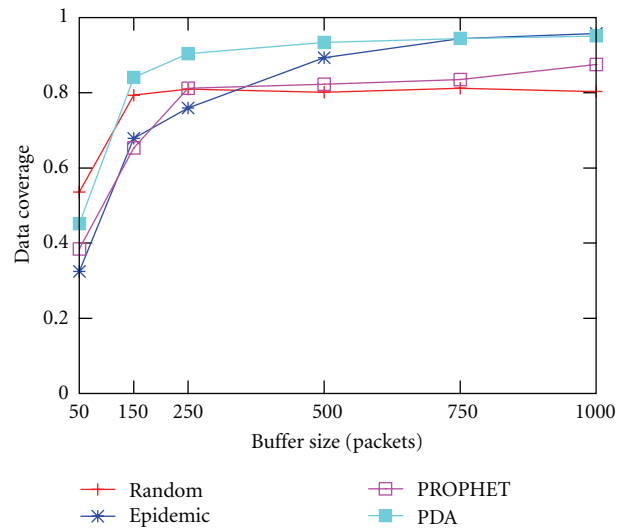



Figure 10: Buffer size versus data coverage.

Epidemic is "cache-hungry", as it would exchange messages whenever nodes encounter other nodes. So many packets would be stored and cached in the buffer. When the buffer is full, a cache replacement is needed for the algorithm. Here we use a FIFO (first-in-first-out) strategy to replace the old packets for all the algorithms. When the buffer size is larger, more packets are stored and exchanged between the nodes, and the data transmissions and data coverage also increase. Yet PDA uses a selective strategy for the data exchange, so it requires relatively less buffer compared with Epidemic, which in return improves its performance.

*5.2.3. Impact of Moving Speed.* The moving speed of nodes reflects the activeness of the network. When nodes move faster, more nodes would meet and make data exchanges. Figures 11 and 12 show that the data transmissions and data coverage increase with the average speed of nodes. When nodes move slowly (e.g., 2 m/s), all algorithms incur small transmissions and small data coverage as there are fewer opportunistic connections. Compared with Epidemic and PROPHET, PDA cuts down about 62% and 47% of message transmissions and increases about 5~8% of the data coverage.

*5.2.4. Comparison of Delay.* Figure 13 shows that besides Random, the data coverage increases quickly with time and goes to a relatively stable point when the time is greater than 600 s. This is because all the algorithms are based on the epidemic copy strategy, and for PDA and PROPHET there is an initial phase when messages are copied among the nodes, which cuts down the delay for data gathering. But in Random, data would not be exchanged between ordinary nodes, so it has larger delay on the data gathering process.

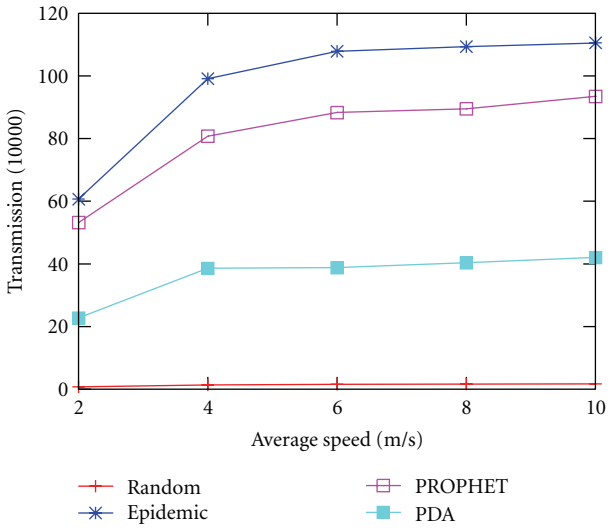



Figure 11: Average speed versus messages.

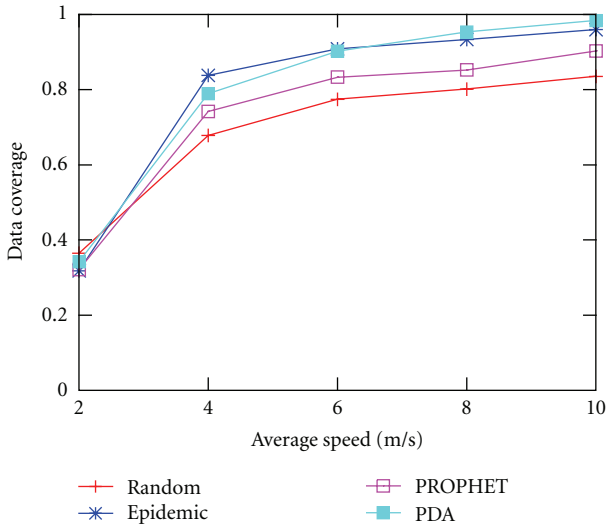



Figure 12: Average speed versus data coverage.

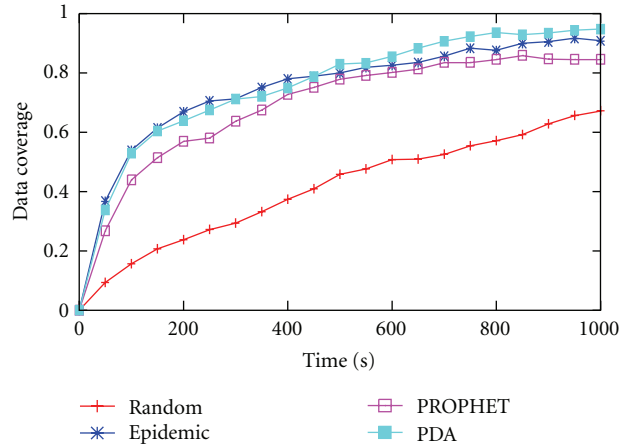



Figure 13: Change of data coverage with time.

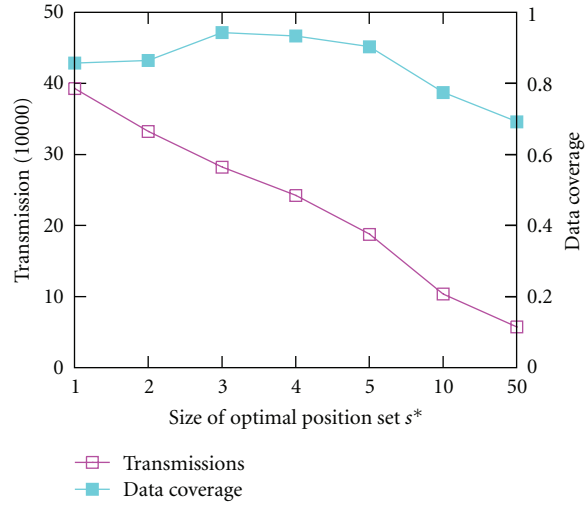



Figure 14: Impact of optimal position set.

Random has a delay of about 500 s for its data coverage to climb up to 60% compared with other algorithms.

### 5.3. Impact of Other Factors

*5.3.1. Impact of Optimal Position Set.* The optimal position $s^*$ could be a point or a set of points. If $s^*$ is a set, then the sink would move among the positions within $s^*$ to collect the sensing data. $s^*$ could be computed as the procedure described in Section 4.3.2 except that there are $m$ ($m = |s^*| > 1$) edges inserted into the graph $G$ to compute the spanning tree with minimal Cost($s$, $G$). Then the sink would visit each positions in $s^*$ accordingly, which is a *feasible route design problem* [4]. As illustrated in Figure 14, when $s^*$ increases, the transmissions goes down, and the data coverage increases when the size of $s^*$ is less than 4 and then decreases when the size grows larger. This is because the sink moves among the positions in $s^*$ for data gathering, and fewer hops are needed to route data from the source to the sink. When the size of $s^*$ is as large as 50, most of the nodes assume that they could communicate with the sink. So they have similar data forward probability and hence suppress the data transmissions among the nodes. Yet the sink could not appear in all the positions at the arranged time, and not every node could communicate with the sink, so the data coverage goes down accordingly. When the size of $s^*$ is 3~4, PDA has a performance with fewer transmissions and higher data coverage (about 94%).

*5.3.2. Impact of Metadata.* PDA collects metadata based on *MTables* within every observed period (or sliding window). The default size of window is 5 epochs, and each epoch is 100 s. As illustrated in Figure 15, the cost of metadata collection takes only a little portion of the whole transmissions for data gathering. When node connection threshold Th is 0.5 s, PDA has the largest cost of metadata collection, and the transmissions are about 7% of the whole transmissions;

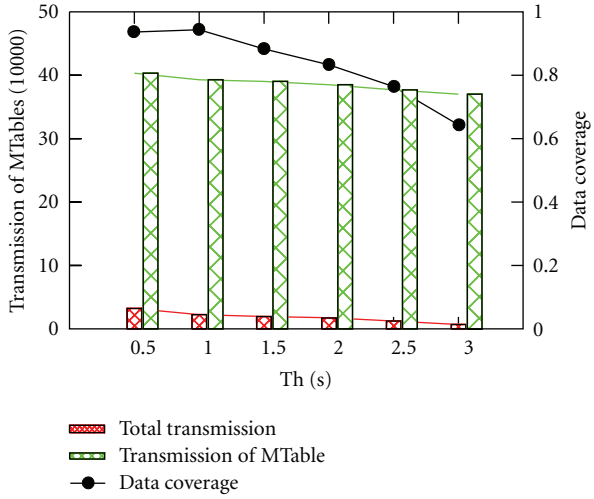



Figure 15: Impact of node connection threshold.

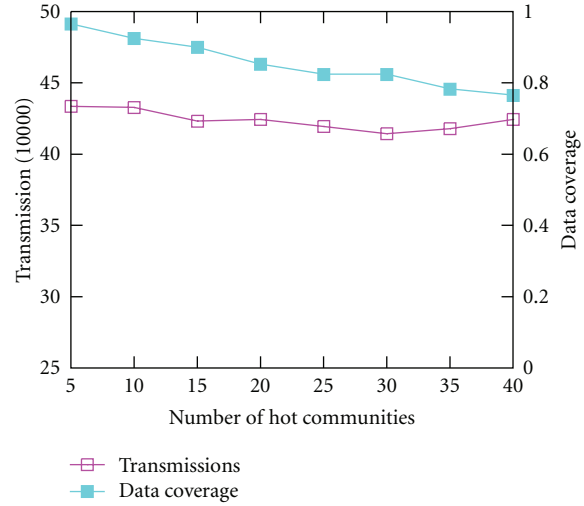



Figure 16: Impact of hot communities.

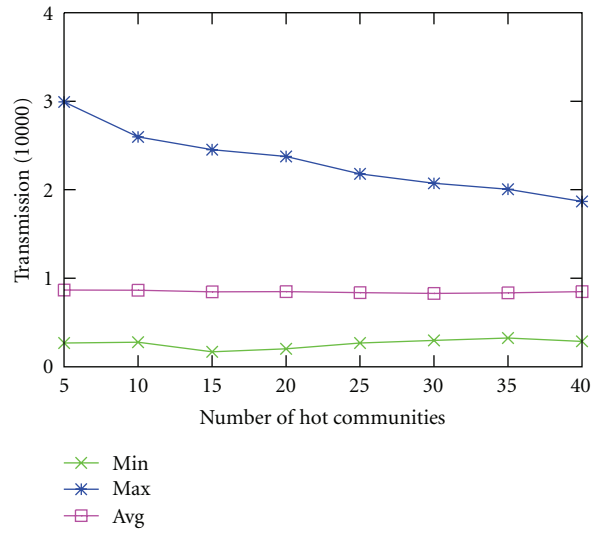



Figure 17: Unbalance of message transmissions.

when Th $>$ 1.0 s, the cost of metadata collection is less than 5%. This is because PDA adopts a controlled-flooding strategy with the time-to-live TTL $=$ 5 and the metadata exchange probability $\gamma = 0.5$. Meanwhile, when node connection threshold increases, the size of *MTable* goes down; this cuts down the transmissions for metadata. Yet larger Th would also affect the precision of metadata and hence affect the prediction of data gathering points. As illustrated in Figure 15, 1.0~1.5 s is a suitable range for the connection threshold, which decreases the cost of metadata collection and does not harm the overall data coverage.

*5.3.3. Impact of Hot Communities.* Figure 16 illustrates the impact of hot communities. As the number of hot communities increases, the data transmissions are relatively the same, yet the data coverage goes down from 94% to 78%. In PDA, hot communities of each node are randomly selected from the hot communities of the network, and each path of the node would contain at least one hot community of the node's hot communities. When there are fewer hot communities in the network, nodes have fewer overlapped hot communities, and the paths of nodes intersect with each other with higher probability, and there are more chances for nodes to meet each other and exchange their sensing data when traveling along their paths. So the pattern of node movements is formed here, and data is routed to the sink at predefined data gathering points and finally increases the data coverage. This also reflects that PDA is more efficient and suitable for cases when nodes move according to the patterns.

In order to study the unbalance of message transmissions in the network, we also present the maximal, minimal, and average number of message transmissions in PDA. As illustrated in Figure 17, the average number of sent messages is about 8000 for each node, and the minimal number is around 3000 messages. However, the maximal number of transmitted messages is about 30,000 when there are only 5 hot communities, and it goes down when there are more hot communities. When there are fewer hot communities (e.g., 5), the nodes are more likely to encounter each other at the communities. These nodes are the active nodes and they would exchange data from each other, making the maximal number of message transmissions larger. Here we assume that the nodes have enough energy to route their own or relayed messages. Yet it is possible that some nodes could not route their messages when they are out of energy. The unbalance of energy consumption is a common problem for the sensor network, and we leave this part of discussion as our future work.

## 6. Conclusion

As the integration of opportunistic networks and wireless sensor networks, data gathering is becoming an important issue in OWSNs. In this paper we have proposed an efficient data gathering algorithm that takes advantage of the pattern

of node movements. It collects the network metadata to create a contact graph, based on which the optimal data gathering positions are calculated and the sink is controlled to move to these positions to collect the sensing data, avoiding lots of unnecessary transmissions. Extensive experimental results show that the proposed algorithm can reduce about 40~60 percent of message transmissions and improve data collection coverage rate about 8~12 percent, compared with other epidemic and probabilistic data gathering algorithms.

For the future work, we are going to implement and deploy PDA on real test bed to validate the performance of the algorithm. We are also planing to investigate the impact of cache strategies and data redundancy of nodes to further improve the performance of the data gathering algorithms in OWSNs.

## Appendix

For the static sensor network, the network topology is fixed. Each node, for example, $a_i$, generates some amount of data $d_i$ ($i = 1, 2, \ldots, n$), which are exchanged among the nodes and finally routed to the sink node $s$. The data gathering problem hence is to minimize the total cost of sending the data set $D = \{d_1, d_2, \ldots, d_n\}$ to the sink node $s$ under the bandwidth constraint. The constraint could be stated as follows: any link in the network should have enough bandwidth to transmit data $d_j$ when the link is connected.

A virtual node vs is then added into the network. vs connects to all the other nodes, and the dotted lines are the virtual edges (as in Figure 18). The sensor network then becomes a flow network, denoted as $G = (V, E)$ with nodes $a_i \in V$, virtual source vs $\in V$, and sink $s \in V$, where edge $(u, v) \in E$ has the capacity $c(u, v) > 0$, flow $f(u, v) \geq 0$, and unit cost $a(u, v) \geq 0$; the cost of sending the flow is $f(u, v) * a(u, v)$. The *data gathering problem* on a static sensor network now becomes a *minimal-cost flow problem (MCF problem)*, which is to minimize the total cost of sending a flow of data $\sum d_i$ from vs to the sink $s$:

$$C = \sum_{(u,e)\in E} a(u, v) * f(u, v) \tag{A.1}$$

with constraints:

capacity constraints:

$$f(u, v) \leq c(u, v), \tag{A.2}$$

skew symmetry:

$$f(u, v) = -f(v, u), \tag{A.3}$$

flow conservation:

$$\sum_{w\in V} f(u, w) = 0, \quad \forall u \neq s, t, \tag{A.4}$$

required flow:

$$f(\text{vs}, a_i) = d_i, \qquad \sum_{w\in V} f(w, s) = \sum d_i. \tag{A.5}$$

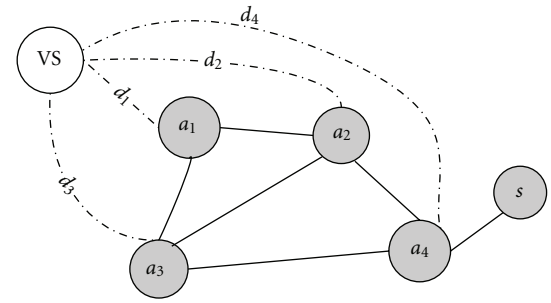



Figure 18: Data are flowed from the virtual source node vs through the sensor network and gathered at the sink node $s$.

The capacity constraint relates to the bandwidth constraint, and the flows from vs to all other nodes are fixed as $f(\text{vs}, a_i) = d_i$. When the cost of sending data from virtual node vs to all the other nodes $a_i$ is subtracted from the total cost $C$ in *MCF*, the result is the minimal cost of data gathering in the fixed network.

## Acknowledgments

This work is supported by the Natural Science Foundation of China (no. 61202012), Natural Science Foundation of Fujian (no. 2011J05156), Fundamental Research Funds for the Central Universities (nos. 2012121030, 2011121049), and Open Project Foundation of the Key Laboratory of Data Engineering and Knowledge Engineering, Ministry of Education (no. KF2011002).

## References

[1] L. J. Chen, C. H. Yu, C. L. Tseng, H. H. Chu, and C. F. Chou, "A content-centric framework for effective data dissemination in opportunistic networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 761–772, 2008.

[2] Y. P. Xiong, L. M. Sun, J. W. Niu, and Y. Liu, "Opportunistic networks," *Journal of Software*, vol. 20, no. 1, pp. 124–137, 2009 (Chinese).

[3] X. Wu and G. Chen, "Dual-Sink: Using mobile and static sinks for lifetime improvement in wireless sensor networks," in *Proceedings of the 16th International Conference on Computer Communications and Networks (ICCCN '07)*, pp. 1297–1302, August 2007.

[4] X. Xu, J. Luo, and Q. Zhang, "Delay tolerant event collection in sensor networks with mobile sink," in *Proceedings of the 29th Conference on Information Communications (INFOCOM '10)*, pp. 2471–2479, March 2010.

[5] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 27–34, ACM, August 2003.

[6] H. Ochiai, H. Ishizuka, Y. Kawakami, and H. Esaki, "A dtn-based sensor data gathering for agricultural applications," *IEEE Sensors Journal*, vol. 11, no. 11, pp. 2861–2868, 2011.

[7] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet," *ACM SIGOPS Operating Systems Review*, vol. 36, no. 5, pp. 96–107, 2002.

[8] R. Ayaki, H. Shimada, and K. Sato, "A proposal of sensor data collection system using mobile relay nodes," *Wireless Sensor Network*, vol. 4, no. 1, pp. 1–7, 2012.

[9] X. Liu, Q. Huang, and Y. Zhang, "Combs, needles, haystacks: balancing push and pull for discovery in large-scale sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 122–133, ACM, November 2004.

[10] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: a tiny aggregation service for Ad-Hoc sensor networks," in *Proceedings of the ACM Symposium on Operating System Design and Implementation (OSDI '02)*, 2002.

[11] S. Burleigh and K. Scott, "Bundle protocol specification," IETF Request for Comments RFC, vol. 5050, 2007.

[12] L. Zhang, X. W. Zhou, J. P. Wang, Y. Deng, and Q. W. Wu, "Routing protocols for delay and disruption tolerant networks," *Journal of Software*, vol. 21, no. 10, pp. 2554–2572, 2010 (Chinese).

[13] A. Vahdat and D. Becker, "Epidemic routing for partially connected Ad hoc networks," Tech. Rep. CS-2000-06, Duke University, 2000.

[14] K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer, "Delay tolerant mobile networks (DTMNs): controlled flooding in sparse mobile networks," in *Proceedings of the 4th International IFIP-TC6 Networking Conference: Networking Technologies, Services, and Protocols, Performance of Computer and Communication Networks, Mobile and Wireless Communications Systems (NETWORKING '05)*, pp. 1180–1192, May 2005.

[15] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking*, pp. 62–66, ACM, June 2007.

[16] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *Lecture Notes in Computer Science*, vol. 3126, pp. 239–254, 2004.

[17] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive routing for intermittently connected mobile ad hoc networks," in *Proceedings of the 6th International Symposium on a World of Wireless Mobile and Multimedia Networks*, pp. 183–189, IEEE, 2005.

[18] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE rap: social-based forwarding in delay tolerant networks," in *Proceedings of the 9th ACM International Symposium on Mobile ad Hoc Networking and Computing*, pp. 241–250, ACM, May 2008.

[19] E. Bulut and B. Szymanski, "Exploiting friendship relations for efficient routing in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2254–2265, 2012.

[20] W. Gao, G. Cao, T. La Porta, and J. Han, "On exploiting transient social contact patterns for data forwarding in delay tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 151–165, 2013.

[21] J. Niu, X. Zhou, Y. Liu, L. Sun, and J. Ma, "A message transmission scheme for community-based opportunistic network," *Journal of Computer Research and Development*, vol. 46, no. 12, pp. 2068–2075, 2009 (Chinese).

[22] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 215–233, 2003.

[23] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 1407–1418, IEEE, March 2005.

[24] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design of sensor networks," in *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks*, pp. 129–145, Springer, 2003.

[25] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*, 1993.